ThreeD.tiff ¬

ThreeD

Scene Modeler and Renderer

Version 0.60 September 1995

by Kieran Jones

Contents

ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

2 Shapes ;ThreeDmanual.rtfd;shapes;¬

2.1 Shape Basics ;ThreeDmanual.rtfd;shapeBasics;¬

2.1.1 Shape Hierarchy ;ThreeDmanual.rtfd;hierarchy;¬

2.1.1.1 Root Shape ;ThreeDmanual.rtfd;rootShape;¬

2.1.2 Creation ;ThreeDmanual.rtfd;shapeCreation;¬

2.1.2.1 Initial Orientation ;ThreeDmanual.rtfd;initialOrientation;¬

2.1.2.2 Shape Naming ;ThreeDmanual.rtfd;shapeNaming;¬

2.1.3 Selection ;ThreeDmanual.rtfd;selection;¬

2.1.3.1 Selecting Single Shapes ;ThreeDmanual.rtfd;selectSingle;¬

2.1.3.2 Selecting All Shapes ;ThreeDmanual.rtfd;selectAll;¬

2.1.3.3 Selected Shape Highlighting ;ThreeDmanual.rtfd;selectHighlight;¬

4 Lights ;ThreeDmanual.rtfd;lights;¬

4.1 Light Intensity ;ThreeDmanual.rtfd;lightIntensity;¬

4.2 Ambient Light ;ThreeDmanual.rtfd;ambientLight;¬

4.3 Distant Light ;ThreeDmanual.rtfd;distantLight;¬

4.4 Spot Light ;ThreeDmanual.rtfd;spotLight;¬

4.4.1 X and Z Angles ;ThreeDmanual.rtfd;xzAngles;¬
4.4.2 Cone Angle ;ThreeDmanual.rtfd;coneAngle;¬

5 Sequencing ;ThreeDmanual.rtfd;sequencing;¬

5.1 Sequence Generation ;ThreeDmanual.rtfd;seqGeneration;¬

5.1.1 Rotation Axis ;ThreeDmanual.rtfd;seqRotationAxis;¬
5.1.2 Rotation Increment ;ThreeDmanual.rtfd;seqRotationInc;¬

7 Saving Scene Data ;ThreeDmanual.rtfd;savingScenes;¬

7.1 Saving Interactive Scenes ;ThreeDmanual.rtfd;savingInteractive;¬

7.2 Saving Photoreal Rendered Frames ;ThreeDmanual.rtfd;savingPhotoreal;¬

7.3 Copying ;ThreeDmanual.rtfd;copying;¬

7.3.2 Photoreal Frame Copies ;ThreeDmanual.rtfd;photorealCopies;¬

8 Reading In Scene Data ;ThreeDmanual.rtfd;readingScenes;¬

8.1 Reading in Interactive Scenes ;ThreeDmanual.rtfd;readingInteractive;¬

8.2 Reading in Photoreal Rendered Frames ;ThreeDmanual.rtfd;readingPhotoreal;¬

8.2.1 Image File Selection ;ThreeDmanual.rtfd;imageSelection;¬

9 Known Problems ;ThreeDmanual.rtfd;knownProblems;¬

10 Legal Notes ;ThreeDmanual.rtfd;legalNotes;¬

ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

intro;¬0 Introduction

ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

ThreeD is a basic three-dimensional scene modeling and rendering program based on NeXT's 3DKit and Pixar's RenderMan. It has the following general capabilities:

· It allows the interactive creation of a 3-D scene consisting of a variety of basic geometric shapes such as cones, cylinders and spheres. The dimensions, position and surface appearance of each shape can be changed at any time.

- Three light sources are available to illuminate the shapes in a scene.
- The camera can be moved about in a variety of ways within a scene to view the latter from any location.
- A scene can be saved to a file at any time and be read in later for further modification. Its RIB code can also be saved to a file.
- A scene can be rendered photorealistically at any time. The result is displayed on-screen and can be saved as a TIFF image file.
- A rudimentary animation capability allows a sequence of frames to be generated and played back, simulating shape motion.

ThreeD was developed by the author as a means of learning NeXTSTEP programming. As a result, it is more a product of accumulation than design. This fact is reflected in the internal structure of the program, which, if the author were to do it again, would be rebuilt from scratch.

ThreeD is therefore not of commercial quality and lacks the features and versatility hoped for in a commercial product. It has never been subjected to extensive testing by aggressive users and almost certainly contains bugs not uncovered by the author.

It is for these reasons that ThreeD is distributed as freeware. As such, it should be used Ð and hopefully enjoyed Ð as an elaborate toy rather than as a serious tool for commercial use. Please see ;ThreeDmanual.rtfd;legalNotes;¬[10 Legal Notes] for the standard disclaimers and distribution conditions.

Users can direct comments by email to the author at:

`kjones@freenet.vancouver.bc.ca`

basics;¬1 Basics
ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

When ThreeD is started it displays the following items:

- A window containing a single view showing the scene as viewed by the camera.
- Two control panels, one on either side of the scene window.
- The main menu.

The following sections describe each of these items.

sceneWindowAndView;¬**1.1 Scene Window and View**

The scene window contains a view that displays the scene as it would appear through the camera lens and in effect acts as the camera's viewfinder. At startup the window is titled "Interactive Scene." Most manipulation of scene elements is done through mouse actions within the view presented in the scene window. Initially, the scene is empty and the view displays only empty world space, the world space axes, some annotation and the rotator circle. These elements are described below.

worldSpace;¬1.1.1 World Space

A scene exists within a coordinate system known as **world space** and is composed of a camera, zero or more shapes and three lights. Each of these elements has its own private coordinate system as well, known as its **local space**.

locTerms;¬1.1.1.1 Location Terminology

The terms location, orientation and position have distinct meanings in ThreeD. **Location** refers to the world coordinates of the origin of the camera or shape. Translations change location. **Orientation** refers to the rotation of the camera or shape relative to some other shape or to the world axes. Rotations change orientation. **Position** represents location and orientation taken together.

If a shape rotates about its own local space origin, its orientation and position may change but its location does not.

Note: ThreeD always indicates locations using world space coordinates.

axisTerms;¬1.1.1.2 Axis Terminology

The names of the world space axes are always given in uppercase X, Y and Z. The names of the camera and shape local space axes are always given in lowercase x, y and z.

worldAxes;¬1.1.1.3 World Space Axes

By default, ThreeD displays the X, Y and Z coordinate axes of world space. They can be turned off and on by a button in the "World Space" control group in the Options subpanel of the Interaction control panel, described in ;ThreeDmanual.rtfd;optionsPanel;¬[1.2.2.3 Options].

The world axes are not labelled, but conform to the following colour-coding scheme used by ThreeD for all axes:

X yellow
Y cyan
Z magenta

The positive end of each world axis is marked with a small cross. Each axis is 0.5 units long in world space and together they look like this:

847615_paste.eps ¬

World Space Axes

In a new, empty scene these axes can be seen in the centre of the scene window. Only the horizontal X and the vertical Y axes are apparent Ð the Z axis is seen on-end, moving into the screen. As the camera moves about the scene the appearance of these axes changes, emulating the change that occurs when a viewpoint in three dimensions changes.

The camera is set at a default viewpoint or **standard position** when a new scene is begun, and it is this position that determines the inital appearance of the world axes. Standard positions are described in ;ThreeDmanual.rtfd;stdPositions;¬[3.1.2 Standard Positions].

cubeGrid;¬1.1.1.4 World Space Cube Grid

The world space cube grid acts as a visual aid to enhance the perception of depth and distance within world space.

It consists of a series of nested wireframe cubes, one inside another, centered around the world origin and extending out from this point equally in all directions.

The cubes are displayed using a blue hue whose saturation increases with distance from the world origin. The innermost cube is always white and the outermost one is always fully-saturated blue. Cubes at intermediate distances have intermediate saturations.

The spatial appearance of the cube grid can be modified using the "World Space" control group in the Options subpanel of the Interaction control panel described in ;ThreeDmanual.rtfd;optionsPanel;¬[1.2.2.3 Options]. By default, ThreeD hides the cube grid.

bgColour;¬1.1.1.5 World Space Background Colour

The background colour of world space as displayed in the scene view can be changed. By default, ThreeD uses opaque black.

Note: Background colour modification as described in this section is only applicable to the scene view in the interactive scene window. A background colour present in this window does not carry over to the scene's rendered photoreal image.

Background colour is modified by first ensuring that no shape is selected in the scene. Shape deselection is described in ;ThreeDmanual.rtfd;deselection;¬[2.1.3.5 Shape Deselection]. If a shape remains selected during the following operation, the shape's colour will be changed rather than the background's.

The "Tools > Colors..." menu item is then clicked to bring up the color panel. The desired colour Ð including opacity if required Ð is set , dragged from the panel and dropped anywhere into the scene view. The scene is then redisplayed with the new colour.

Reducing background colour opacity also reduces the opacity of the annotation text if the latter is being displayed. ;ThreeDmanual.rtfd;annotation;¬[1.1.2 Annotation] describes ThreeD annotation.

Note that the default opaque black of a scene background is a true opaque colour. By contrast, the visually black background of a photoreal rendered image is not black, but transparent.

As a result, a scene copied from the ThreeD scene window and pasted into another program such as Edit will show a true, black background as expected. However, a rendered image copied from the ThreeD image window and pasted in the same manner will retain its transparent background and show whatever opaque colour exists behind it.

annotation;¬1.1.2 Annotation

Annotation refers to the two lines of text running along the top and bottom of the scene view. These lines are used to display certain scene parameters that change over time.

The top line always displays camera status, including the latter's standard position or the type of motion it is currently performing, if any.

The bottom line always displays the changing world X, Y and Z coordinates of the camera or shape when being translated, or the static coordinates of the camera when nothing is being translated. The display doesn't change for rotations that leave camera or shape location unchanged.

Annotation can be turned on and off, or set transparent or opaque by a control in the Interaction control panel, described in ;ThreeDmanual.rtfd;optionsPanel;¬[1.2.2.3 Options].

rotator;¬1.1.3 Rotator Circle

By default, ThreeD displays a yellow circle surrounding most of the scene, centred on the midpoint of the scene view with a diameter equal to the view's width or height, whichever is smaller.

This **rotator circle** is a visual cue indicating what kind of rotation will occur when the mouse is used to apply a rotation to the camera or to a shape. Shape rotation is described in ;ThreeDmanual.rtfd;shapeRot;¬[2.3.5 Rotation] and camera rotation in ;ThreeDmanual.rtfd;cameraRot;¬[3.2.3 Rotation]. By default, while the mouse is dragged within the circle, rotation can potentially occur about any of the three axes x, y and z. While dragging within the smaller corner areas lying outside the circle, rotation can only occur about the z-axis.

Further restriction of rotation to specific axes is described in ;ThreeDmanual.rtfd;motionPanel;¬[1.2.2.1 Motion].

Rotator circle visibility can be turned on and off by a button in the Interaction control panel, also described in ;ThreeDmanual.rtfd;motionPanel;¬[1.2.2.1 Motion].

displaying;¬1.1.4 Displaying

The scene window can be resized, miniaturized and closed in the usual manner. When the window is resized, the rotator circle adjusts its diameter accordingly.

Closing the scene window does NOT delete the scene displayed within it Ð it merely removes the window from the screen. A closed scene window can be re-opened at any time by clicking the "Tools > Scene Window" menu item.

controlPanels;¬1.2 Control Panels

To the left and right of the scene window is a control panel, each containing various controls used in scene creation and modification. The left panel is titled "Setup" and the right panel is titled "Interaction". At the top of each is a popup list used to select and display subpanels containing controls grouped by function. These are described in the following sections.

setup;¬1.2.1 Setup

The Setup control panel contains controls for setting up the basic elements of the scene. This panel can be miniaturized and closed in the usual fashion. If closed, it can be re-opened at any time by clicking the "Tools > Setup..." menu item.

The Setup panel's popup list is used to select Shapes, Lights or Camera setups which are described next.

shapePanel;¬1.2.1.1   Shapes

The Shapes subpanel contains a pulldown list named "Types" and a slider named "Resolution".

The **Types** pulldown list is used to create new shapes in the scene. Nine different shape types can be created by selecting from the list. Shape creation is described in ;ThreeDmanual.rtfd;shapeCreation;¬[2.1.2 Creation].

The **Resolution** slider is used to change the resolution of displayed shape surfaces in the scene. Resolution can be varied in integer amounts from a low of 3 to a high of   20 with a default of 8. The scene is continuously redisplayed as resolution is changed.

This resolution is applied globally Ð all shapes receive the same resolution Ð and only affects the scene displayed in the scene window. It is not applied to a photoreal rendering of a scene. The latter is performed at the dpi resolution requested specifically for photoreal rendering as described in ;ThreeDmanual.rtfd;renderPanel;¬[6.3.1 Render Panel].

Note: Higher resolutions require more processing time and take longer to redisplay. The slider's range is divided into blue, yellow and red sections to identify low, medium and high values. Caution should be exercised when specifying high resolutions. This is discussed in item 2 of ;ThreeDmanual.rtfd;knownProblems;¬[9 Known Problems].

lightsPanel;¬1.2.1.2 Lights

The Lights subpanel contains sliders to vary the intensity of the ambient, distant and spot lights available to the scene and sliders to vary angles specific to the spot light. Lights are described in ;ThreeDmanual.rtfd;lights;¬[4 Lights].

cameraPanel;¬1.2.1.3 Camera

The camera subpanel contains a pulldown list named "Positions" and a text field named "Distance Away". Both controls are used to set the camera's standard position. Seven different standard positions can be set by selecting from the list. Standard positions are described in ;ThreeDmanual.rtfd;stdPositions;¬[3.1.2 Standard Positions].

interaction;¬1.2.2 Interaction

The Interaction control panel contains controls for interacting with scene elements. This panel can be miniaturized and closed in the usual fashion. If closed, it can be re-opened at any time by clicking the "Tools > Interaction..." menu item.

The Interaction panel's popup list is used to select Motion, Sequencing or Options interactions which are described next.

motionPanel;¬1.2.2.1 Motion

The Motion subpanel contains control groups named "Rotator", "Flight", and

"Look At" and a slider named "Drag Sensitivity".

The **Rotator** group contains a set of radio buttons to used to select which rotation axes will be active during a shape or camera rotation. All seven combinations of axes are available Ð single, paired and all three. The setting can be changed at any time and remains in effect for all subsequent rotations until changed again.

The value set here overrides the action of the rotator circle described in ;ThreeDmanual.rtfd;rotator;¬[1.1.3 Rotator Circle]. If a combination is selected that doesn't include the z-axis, no z-axis rotation will occur even if dragging is done within the z-axis area outside the circle. In such a case there will be no rotation about any axis.

Also included in the rotator group is a button used to set rotator circle visibility. Clicking on it alternately hides and shows the circle. By default, ThreeD shows the rotator circle.

The control action delineated by the circle remains in effect at all times, regardless of whether the circle is visible or not.

The **Flight** group contains controls to set and change camera flight direction and speed and are fully described in ;ThreeDmanual.rtfd;flight;¬[3.2.4 Flight].

The **Look At** group contains controls to force the camera to look at a specific shape or at the world origin, overriding the camera's current orientation. This capability is described in ;ThreeDmanual.rtfd;lookAt;¬[3.2.5 Look At].

Finally, the **Drag Sensitivity** slider is used to change the responsiveness of any action that is controlled by dragging the mouse within the scene. Smaller or larger increments of change for the same mouse movement can be obtained by setting finer or coarser drag sensitivities with the slider.

Finer settings are useful when precise, small-scale control over motion is desired Ð when working very close to shapes, for example. Coarser settings are useful when large mouse inputs are desired Ð for example, when long distances must be covered quickly.

sequencingPanel;¬1.2.2.2   Sequencing

The Sequencing subpanel contains a slider named "# Frames to Generate" and a button named "Play Once Interactively". These controls are used to generate and display a sequence of scene frames which forms the basis of ThreeD's rudimentary animation capability.

Sequencing is described in ;ThreeDmanual.rtfd;sequencing;¬[5 Sequencing].

optionsPanel;¬1.2.2.3   Options

The Options subpanel contains a control group named "Shapes During Interaction", a control group named "World Space" and a set of radio buttons named "Annotation".

The **Shapes During Interaction** group contains controls used to determine how shapes will appear when displayed during interaction, described in ;ThreeDmanual.rtfd;appearance;¬[2.1.6 Appearance]; a check box used to control display of shape dimensions, described in ;ThreeDmanual.rtfd;handles;¬[2.3.2 Edit Handles]; and a check box used to turn auto-rotation on and off, described in ;ThreeDmanual.rtfd;autoShapeRotation;¬[5.3 Auto Shape Rotation].

The **World Space** group contains a button used to hide and show the world space axes and a subgroup named "Cube Grid" to control the appearance of the world space cube grid. World space axes are decribed in ;ThreeDmanual.rtfd;worldAxes;¬[1.1.1.3 World Space Axes] and the cube grid is described in ;ThreeDmanual.rtfd;cubeGrid;¬[1.1.1.4 World Space Cube Grid].

Clicking on the world space axes button alternately hides and shows them and can be done at any time. Only axis visibility is affected Ð world space itself is not altered. By default, ThreeD shows the world space axes.

Cube grid modification is accomplished by a button and two sliders in the "Cube Grid" subgroup.

The button is alternately clicked to show and hide the cube grid. By default, ThreeD hides the cube grid.

The slider named "Spacing" is used to vary the spacing between cubes from fine to coarse. Finer spacings produce a greater number of cubes which are more closely spaced. Coarser spacings produce fewer cubes which are more widely spaced.

The slider named "Farthest Corner" is used to vary how far away the outermost cube is from the world origin, from near to far. This cube contains all the other cubes in the grid and sets the limit on the grid's extent in world space.

The parameters of the cube grid can be changed at any time. Only the grid's appearance is affected Ð world space is not altered.

Note: Showing the cube grid with fine spacings and far farthest-corners can slow down scene redisplay.

The **Annotation** radio buttons are used to set the appearance of the annotation displays along the top and bottom of the scene view. Annotation text background can be set to opaque black, blocking out the scene behind it. This guarantees that annotation remains visible regardless of the colour of any shapes present behind the text. This is the default setting.

The background can also be set transparent, allowing shapes behind to show through the text. Annotation can also be turned off altogether.

Note: Using a scene background colour having less than full opacity also causes the annotation text itself to become less opaque and harder to see as a result. Setting the scene background colour is described in ;ThreeDmanual.rtfd;bgColour;¬[1.1.1.5 World Space Background Colour].

mainMenu;¬**1.3 Main Menu**

The main menu of ThreeD contains five submenus and the standard "Hide" and "Quit" items. In addition to the standard "Info" and "Windows" submenus, there are three submenus containing customized items specific to ThreeD. These submenus and the functions of each of their items are described in the following sections.

fileMenu;¬1.3.1 File

327560_paste.eps ¬

editMenu;¬1.3.2 Edit

964553_paste.eps ¬

toolsMenu;¬1.3.3 Tools

937576_paste.eps ¬

assistance;¬**1.4 On-line Assistance**

Two sources of on-line assistance are available when using ThreeD: this reference manual and the Help panel. The following sections describe these two sources of information.

refManual;¬1.4.1 Reference Manual

This reference manual constitutes the principal source of information about ThreeD.

It is contained in a file called `ThreeDmanual.rtfd`. This manual file and the actual `ThreeD.app` application file are both contained in a compressed tar file called `ThreeD.tar.Z` which is used to distribute ThreeD to users.

The manual file is intended to be viewed on-screen and/or printed out for reference using the Edit application. It takes up approximately 1MB of disk space and is about 60 single-sided pages in length.

The manual is arranged into numbered sections and subsections, illustrated with figures and tables. A table of contents is included at the beginning but there is no index.

Because the manual was prepared using the Edit application, it is not properly paginated. When printed, page breaks may occur at inappropriate locations. This can happen even in the middle of illustrations, causing the latter to be printed across pages.

Help-type links indicated by blue diamonds are present for each section in the table of contents. Clicking on a section's link diamond when using Edit to view the manual causes the display to scroll to the beginning of that section.

Links are also present throughout the manual wherever cross-references are made to other sections. Clicking on the link diamonds of such references causes the relevant section to be displayed.

Note: The help links have been set up to work correctly when the manual file is named `ThreeDmanual.rtfd`. If this file is renamed, the links will still work, but will first issue a "File system error" message every time they are used, warning that `ThreeDmanual.rtfd` could not be found. Clicking the "OK" button will complete the link successfully within the renamed file.

helpPanel;¬1.4.2 Help Panel

The NeXT Help panel is also available by clicking on the "Info > Help..." menu item. The panel can then be used to display an introduction to this reference manual as well as summary descriptions of ThreeD shape and camera operations.

No Help panel index is available.

shapes;¬2 Shapes

ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

Shapes form the principal elements of a scene.   ThreeD enables shapes to be created, inspected and modified interactively. The following sections describe these operations in detail.

shapeBasics;¬2.1   Shape Basics

Basic shape concepts consist of the shape hierarchy and the operations that can be performed on shapes, including creation, selection, duplication, deletion and control of appearance during interaction. The following sections describe these basic concepts and operations.

hierarchy;¬2.1.1   Shape Hierarchy

All shapes in a scene are related to one another in a hierarchical manner which takes the form of a family tree.

At the top of this family or **shape tree** is a single shape followed by zero or more **descendant** shapes. A shape having at least one descendant is called the **ancestor** of those descendants. Descendants sharing a common direct ancestor are said to be **peers** of each other. Each descendant can have descendants of its own and each of these can have peers of their own and so on, building up a hierarchy of descendant subtrees all connected to the common ancestor at the top.

Moving or scaling a shape causes all of its descendants to be moved or scaled along with it. This is the principal advantage of grouping shapes together in a shape tree Ð motion need only be applied to one shape for its entire descendant group to follow automatically.

A scene's shape tree can be examined using the shape hierarchy browser described in ;ThreeDmanual.rtfd;hierarchyBrowser;¬[2.2.1 Shape Tree Hierarchy Browser]. A shape tree is automatically added to and subtracted from when shapes are created and deleted within the scene. Explicit rearrangement of the shape tree is accomplished by shape grouping, which is described in ;ThreeDmanual.rtfd;grouping;¬[2.3.10 Grouping].

A ThreeD scene has only one shape tree.

Shapes exist both in space and in the shape tree. It is useful to keep in mind that a shape's *spatial* relationship to others around it in world space is independent of its *hierarchical* relationship to others around it in the shape tree. Two shapes can be directly related to each other in the shape tree and be very distant from one another in space. Conversely, two shapes having no direct relationship in the shape tree can be sitting on top of each other in space.

rootShape;¬2.1.1.1 Root Shape

A ThreeD scene always has a top-level shape which acts as the ancestor to all shapes created by the user. ThreeD creates this **root shape** automatically for each scene and names it "root". It cannot be created, duplicated or deleted by user action but it can be selected, moved and scaled like any other shape.

The root shape has no "physical" surface appearance and is never visible within a scene. A new, empty scene always contains the root shape Ð the scene is empty in the sense that it contains no user-created shapes.

All user-created shapes are set to be descended from this root shape and no user-created shape can be made a peer or ancestor of the root shape.

Applying a common motion to all shapes in a scene can be conveniently accomplished by moving the root shape Ð all other shapes in the scene will follow automatically since all are descended from it.

Specific shape operations concerning the root shape are discussed in the relevant sections below.

shapeCreation;¬2.1.2 Creation

Shapes in ThreeD are created by activating the Types pulldown list on the Setup panel's "Shapes" subpanel and selecting a type from the list. The list offers the following nine shapes from which to choose:

- Cone
- Cylinder
- Sphere
- Torus
- Paraboloid
- Hyperboloid
- Polygon
- Patch
- Teapot

When a shape type has been chosen, the following operations are performed:

- a shape of the type chosen is created having the type's default initial appearance.
- if there is no currently selected shape, the new shape is made a direct descendant of the root shape.
- if there is a currently selected shape, the new shape is made its peer.
- the new shape is placed in the scene and is set to be the currently selected

shape.

All newly created shapes are placed at the same initial location in world space with their local origins coincident with the world origin Ð they are placed "dead centre" in the scene.

When a shape is created, ThreeD automatically makes it the selected shape and displays it as such. Manually setting a shape to be the selected shape is discussed in ;ThreeDmanual.rtfd;selection;¬[2.1.3 Selection].

Note: The root shape cannot be created by user action.

initialOrientation;¬2.1.2.1 Initial Orientation

A new shape's initial orientation with respect to the world space axes is set to make the particular geometry of its type as obvious as possible. With one exception ThreeD does this by rotating the shape about its x-axis until its z-axis is vertical and its y-axis points into the screen.

This in effect swaps the orientations of the y and z axes so that the shape's local coordinate axes are no longer coincident with those of world space but are instead rotated ninety degrees out of alignment in the world YZ plane.

The Polygon is the single exception Ð it is not rotated by ThreeD and its initial orientation is coincident with the world space axes.

In all cases the initial orientation of a new shape's x-axis is horizontal and coincident with the world X-axis.

shapeNaming;¬2.1.2.2 Shape Naming

ThreeD assigns a unique name to each shape created or duplicated. Shape names are used to identify shapes in the shape hierarchy browser, described in ;ThreeDmanual.rtfd;hierarchyBrowser;¬[2.2.1 Shape Tree Hierarchy Browser]. They cannot be specified or modified by the user.

A shape name consists of the shape's type followed by a unique integer. The first shape of each given type is assigned the integer 1 and subsequent shapes of the same type are assigned integers incremented by one each time.

For example, the first Cone created in the scene is named "cone1" and the first Hyperboloid is named "hyperboloid1". The next Cone is named "cone2" and so on.

When a single shape is deleted its name is not reused in that scene, so it is possible to have a shape name whose number is greater than the actual number of shapes of that type in the scene. However, if all shapes are selected at once and deleted together in a single operation, shape naming begins again at 1.

selection;¬2.1.3 Selection

A shape must first be selected before it can be operated on. Either a single shape or all shapes can be selected Ð selection of an intermediate number of shapes is not supported.

selectSingle;¬2.1.3.1 Selecting Single Shapes

Selecting a single shape (other than the root shape) is done either by clicking anywhere on its surface or by clicking on its name in the shape hierarchy browser. The latter method is described in ;ThreeDmanual.rtfd;inspectorSelection;¬[2.2.1.1 Shape Selection]. The click-on-shape method is the more commonly used of the two.

Note: ThreeD sometimes has problems determining which shape to highlight as selected. When shapes overlap at the point where clicking to select the shape "on top", ThreeD will sometimes err and set one of the shapes behind as selected instead.

There are two ways around this defect if it occurs. One way is to click on the desired shape at a point where there are no shapes behind it. This may require temporarily moving the camera to a different position which shows such a point.

The other approach is to select the shape from within the shape hierarchy browser.

selectAll;¬2.1.3.2 Selecting All Shapes

Selection of all shapes in the scene is accomplished by clicking on the "Edit > Select All" menu item or entering the Command-a keyboard equivalent.

The selection of all shapes actually causes the root shape to be selected, but because all user-created shapes are descended from the root shape, selecting the latter in effect selects all shapes in the scene.

The root shape can also be directly selected explicitly as a shape, although the effect is identical to selecting all shapes as described above. Since it has no visible surface to click on, the root shape can only be directly selected using the shape hierarchy browser as described in ;ThreeDmanual.rtfd;inspectorSelection;¬[2.2.1.1 Shape Selection].

selectHighlight;¬2.1.3.3 Selected Shape Highlighting

Once a shape is selected, ThreeD highlights it by displaying it with a wireframe appearance and drawing its edit handles and the axes of its local coordinate system. Edit handles are described in ;ThreeDmanual.rtfd;handles;¬[2.3.2 Edit Handles].

Note: The root shape's lack of a visible surface means that when selected, it cannot be displayed with a wireframe appearance Ð only its axes and edit handles will be displayed.

If a non-root shape is being displayed as wireframe with axes and edit handles, it is by definition the selected shape. Any non-root ancestors of the selected shape are also displayed as wireframe to indicate what shapes it descends from, but only the selected shape will have axes and edit handles as well.

selectAxes;¬2.1.3.4 Selected Shape Axes

The local axes of a selected shape are coloured according to the convention ThreeD uses for all axes:

X yellow
Y cyan
Z magenta

In addition, local shape axes are labelled at their positive ends with the letters "x", "y" and "z". This differs from the axes for world space, which are not labelled but use small crosses at their positive ends instead. In addition, world axes are always shorter than shape axes unless the shape has been rescaled to a smaller size.

As previously described, a newly created shape is placed initially at the world origin. This causes the shape's local axes to overlap the world's axes and obscure the latter if they are being displayed. Changing the shape's position will cause its axes to move off of the world axes, causing the latter to become visible.

deselection;¬2.1.3.5 Shape Deselection

A shape is deselected either by clicking anywhere in the black background of world space or by deselecting its name in the shape hierarchy browser. The latter method is decribed in ;ThreeDmanual.rtfd;inspectorDeselection;¬[2.2.1.2 Shape Deselection]. ThreeD redisplays the deselected shape with its solid surface appearance minus the axes and edit handles.

Changing the selected shape is done by clicking on the new shape to be selected. ThreeD deselects the old shape and highlights the new one as described above.

duplication;¬2.1.4 Duplication

A shape is duplicated by first selecting it and then either clicking on the "Edit > Shapes > Duplicate" menu item or entering the Command-d keyboard equivalent. These commands are disabled when no shape is selected.

The newly created duplicate is made a peer of the original and placed in the scene at a position which is offset by a small amount to the upper right of the original. Note that this offset is relative to the original's local axes and will not necessarily correspond to an upper right offset in world space.

Finally, the original is deselected and the duplicate is set as the selected shape.

Note: The root shape cannot be duplicated.

deletion;¬2.1.5 Deletion

To delete a shape it must first be selected and since only one or all shapes can be selected, only one or all shapes can be deleted.

Note: The root shape cannot be deleted.

deleteSingle;¬2.1.5.1 Deleting Single Shapes

A single shape is deleted by first selecting it and then either clicking on the "Edit > Shapes > Remove" menu item or entering the Command-r keyboard equivalent. These commands are disabled when no shape is selected. ThreeD does not warn or prompt for confirmation before deleting single shapes.

The scene is redisplayed with the shape removed and if it had a non-root ancestor this is set as the selected shape.

When a shape is deleted, only it is affected Ð any descendants and peers it had remain untouched.   The shape tree is adjusted to reflect the deletion by moving the shape's descendant subtrees up one level, making them direct descendants of the deleted shape's direct ancestor.

deleteAll;¬2.1.5.2   Deleting All Shapes

All shapes in a scene can be deleted in one operation by first selecting all shapes (in reality the root shape) and then issuing the same "Edit > Shapes > Remove" or Command-r commands used for deleting a single shape.

ThreeD displays a warning and requests confirmation before deleting all shapes in a scene.

Note that although it is actually the root shape that is selected when deleting all shapes only its descendants are actually removed. The root shape remains untouched and is never deleted along with them.

appearance;¬2.1.6 Appearance

While shape or camera motion is occurring or while shape dimensions are being modified, ThreeD's default action is to display all shapes in the scene as wireframe versions of their actual-shape forms. This enables faster redisplay of the scene during the interactive operation while still preserving the shape's geometric form.

This default behaviour can be changed using the "Shapes During Interaction" controls located in the "Options" subpanel of the Interaction control panel. Both form and surface appearance can be set independently of one another.

form;¬2.1.6.1 Form

The form of shape appearance during interaction can be set using the radio buttons named "Display As". The form can be set to either "Actual Shape" (the default) or "Bounding Box".

Actual-shape displays a shape with its true geometry Ð cones look like cones, spheres look like spheres and so on.

Bounding-box displays a shape as a three-dimensional box whose dimensions enclose the shape's extreme boundaries. The volume of a shape's bounding box is almost always greater than the volume of the shape itself Ð sometimes it is much greater.

Displaying shapes as bounding boxes is faster than using the actual shapes, making this setting useful in speeding up scene redisplay when many shapes are present. However, bounding-box display can sometimes be visually confusing, since shapes lose their distinguishing forms and the scene appears as a collection of boxes only.

surface;¬2.1.6.2 Surface

The surface of shape appearance during interaction can be set using the radio buttons named "Using". The surface can be set to either "Wireframe" (the default) or "Solid-Surface".

Wireframe displays shapes using a framework of lines to approximate their surfaces. Scene redisplay is faster with wireframe than with solid-surface. If a bounding-box form is used with wireframe, only the box's edges will be shown. Scene redisplay will be fastest using this combination.

Solid-surface uses solid surface areas to approximate shape appearance. This offers the highest fidelity of shape display but is slower than using wireframe. If a bounding-box form is used with solid surface, the box's sides will be shown as solid. The scene will appear as a collection of solid boxes of different size, shape and colour.

Note: ThreeD always displays the selected shape using wireframe regardless of the wireframe / solid-surface setting.

resolution;¬2.1.6.3 Resolution

The shape resolution setting described in ;ThreeDmanual.rtfd;shapePanel;¬[1.2.1.1 Shapes] affects shape appearance when actual-shape form is used with either wireframe or solid-surface. The bounding-box form is not affected by the resolution setting.

shapeInspector;¬2.2 Shape Inspector

The Shape Inspector is a panel displaying information about a selected shape in the scene. By default, ThreeD does not show this panel. It can be displayed either by clicking on the "Tools > Shape Inspector..." menu item or by entering the Command-i keyboard equivalent.

The Shape Inspector contains a shape tree hierarchy browser, the shape's world coordinates, a splitview showing information about the shape's geometry and its shaders, and a list of available system shaders. The following sections describe each of these components.

hierarchyBrowser;¬2.2.1 Shape Tree Hierarchy Browser

The Shape Inspector contains a three-column browser named "Shape Tree Hierarchy" that displays the current state of ThreeD's shape tree hierarchy. The shape tree concept is described in ;ThreeDmanual.rtfd;hierarchy;¬[2.1.1 Shape Hierarchy].

Shapes are displayed in the browser using the unique names generated for them by ThreeD as described in ;ThreeDmanual.rtfd;shapeNaming;¬[2.1.2.2 Shape Naming]. The root shape, which exists in every scene, is always displayed as "root" in the first, left-most column of the browser. Since no user-created shape can be made a peer or ancestor of the root shape, the first column will never contain anything but the "root" entry.

Subsequent columns in the hierarchy browser will contain varying numbers of shape entries reflecting the state of the shape tree at any given point in time. The browser columns are updated each time a shape is created, duplicated, deleted or regrouped in the scene, with columns being added, deleted and horizontally scrolled as necessary.

Moving to the right within the browser represents a movement down within the shape tree. All shapes in the same column are peers of one another and are direct descendants of the highlighted shape in the column immediately to the left.

A shape having descendants will have the usual ">" browser symbol as part of its entry, pointing to the right to indicate the presence of descendants one level down in the hierarchy from that point.

inspectorSelection;¬2.2.1.1 Shape Selection

Whenever a shape is user-selected by clicking on it within the scene view it is automatically selected in the browser as well. The path down through the shape tree to the selected shape is traced out in the browser by highlighting each of the shape's ancestors, beginning with the root shape. The final shape highlighted in the browser is the selected shape itself.

A corresponding tracing out is done in the scene view by highlighting each non-root ancestor using wireframe. This is described further in ;ThreeDmanual.rtfd;selectHighlight;¬[2.1.3.3 Selected Shape Highlighting].

A single shape can also be user-selected using the browser in the same way a file is selected using the Workspace Manager file viewer. Clicking on a browser shape entry causes that shape to be highlighted in the browser and selected in the scene view, with non-root ancestors being highlighted as described above.

The browser allows only one shape at a time to be selected.

inspectorDeselection;¬2.2.1.2 Shape Deselection

A single shape can be user-deselected using the browser in the same way a file is deselected using the Workspace Manager file viewer. Clicking on the selected shape browser entry while holding down the Shift key causes that shape to be deselected both in the browser and in the scene view.

lostShapes;¬2.2.1.3 Lost Shapes

There are times when a shape is known to exist in a scene but cannot be found. This can occur when the shape is not in the camera's field of view and it is not known where the camera should be pointed in order to see the shape. The shape cannot be selected by clicking on it since it must first be visible before selection can occur. In effect, the shape has been "lost".

Shape selection via the browser can be used to find the shape again if the latter's unique name is known. The shape is first selected in the browser and then looked at using ThreeD's "Look At" function, described in ;ThreeDmanual.rtfd;lookAt;¬[3.2.5 Look At].

worldCoords;¬2.2.2 World Coordinates

The Shape Inspector contains three textfields in a group named "World

Coordinates". The selected shape's X, Y and Z coordinates in world space are always displayed here and are updated whenever the shape's location is changed.

geomInfo;¬2.2.3 Geometry Information

The upper half of the Shape Inspector's splitview contains information about the geometry of the selected shape. The shape's unique name is always displayed first, followed by the names and values of its geometric parameters such as Radius, Zmin, Zmax and so forth.

For the most part the parameter names shown correspond to the same-named parameters modified by the shape's geometry edit handles, described in ;ThreeDmanual.rtfd;geomHandles;¬[2.3.7 Geometry Edit Handles]. One exception to this is the parameter named "ThetaMax", which refers to the shape's wrap angle.

A shape's geometry values are updated in the splitview whenever they are changed in the scene via the edit handles. Each time the selected shape is changed the parameters of the newly selected shape are displayed. Note that when the root shape is selected only its name is displayed since it has no viewable surface geometry.

The Shape Inspector's splitview is display-only and cannot be modified. However, text selections can be made from it in the usual manner for copying and pasting elsewhere if desired.

shaderInfo;¬2.2.4 Shader Information

The lower half of the Shape Inspector's splitview contains information about the shader of the selected shape. To the right is a scrollable list browser named "System Shaders" which displays all the standard shaders available on the system. Directly below the splitview is a row of radio buttons, each of which corresponds to one of the six standard shader types. These shader-specific components are described in ;ThreeDmanual.rtfd;shaders;¬[2.3.9 Shaders].

shapeMod;¬2.3 Shape Modification

Once created, a shape can be modified in a number of ways. It can be translated, scaled, rotated and have its geometric dimensions changed. In addition, its colour and surface appearance can be modified and its position in the shape tree can be altered.

The sections below describe the operations involved in performing shape modification.

modConv;¬2.3.1 Conventions

All shape modification occurs within the shape's local coordinate space, relative to the orientation of its local coordinate axes. As with any shape operation, a shape must first be selected before it can be modified.

For all linear changes along shape axes, the following conventions are adhered to:

- x-axis change:
  - towards positive end of axis — move cursor horizontally to right
  - towards negative end of axis — move cursor horizontally to left
- y-axis change:
  - towards positive end of axis — move cursor vertically upwards
  - towards negative end of axis — move cursor vertically downwards
- z-axis change — same as y-axis but with Alternate key down
- constrain change to one axis — move cursor with Command key down

By following these conventions, linear changes can be applied in either the xy or the xz plane along both axes simultaneously or constrained to only one of them.

Constraint is discussed in the next section.

changeConstraint;¬2.3.1.1 Change Constraint

Change constraint is useful in situations where a change is normally allowed to take place along both axes of the xy or xz plane. In such cases, the change can Ð in theory Ð be applied to only one or the other axis if the mouse is moved very carefully in a purely horizontal or vertical direction.

However, mouse movement of this accuracy is difficult to achieve, making single-axis changes equally difficult to realize.To avoid this problem, ThreeD can be made to constrain a change to a single axis by simply ignoring any mouse movement along the other one. This constraint feature guarantees that the change will be applied to the single desired axis only, regardless of any mouse movement along the other axis.

In use, ThreeD deems the constrained axis to be the one associated with the larger initial displacement when the mouse starts to move with the Command key held down.

handles;¬2.3.2 Edit Handles

A selected shape is displayed with **edit handles** Ð small coloured squares that allow shape location, scale and geometry to be modified. Each handle on a shape controls a different shape parameter. To change a parameter's value, mousedown is perfomed on the handle controlling the parameter and the mouse is dragged.

By default, the parameter value is displayed as it changes during the mouse drag. This display can be turned on and off using the checkbox control named "Show Values When Handles Used" on the Interaction panel's Options subpanel. Note that not all edit handles show parameter values.

Edit handles do not follow the cursor as it is dragged. They serve only as indicators marking the spot at which mousedown should occur to begin the parameter change.

All shapes possess a white translation handle at their origin and a cyan scaling handle located a short distance away. All remaining handles control geometry parameters specific to the shape type and are yellow in colour. Their relative positions indicate which parameter they vary.

In most cases, yellow geometry handles respond only to horizontal cursor motion and ignore vertical cursor motion. To increase such geometry values, the cursor is dragged to the right; to decrease, it is dragged to the left.   Exceptions to this general rule are noted below for the relevant shape types. The Alternate key has no effect on geometry handles.

Like all dragging operations, edit handle dragging is affected by the drag sensitivity setting as described in ;ThreeDmanual.rtfd;motionPanel;¬[1.2.2.1 Motion].

shapeTrans;¬2.3.3   Translation

Shape translation is accomplished by selecting the shape, mousedowning on its white translation handle and dragging until it reaches the desired location, constraining to one axis if desired.

During translation the shape's world X, Y and Z coordinates are displayed in the bottom annotation line as they change. Once shape motion ceases and mouseup occurs, the camera world coordinates are redisplayed.

polyPatchTrans;¬2.3.3.1 Polygon and Patch Translation

The Polygon and Patch shapes have translation handles that possess an additional capability. As Polygon vertices or Patch control points are moved about, they can become significantly displaced from the shape's local origin. In extreme cases the shape's entire surface will be shifted away from its displayed axes.

To correct this possibly misleading appearance, the vertices or control points can be shifted back as a group about the origin again by clicking on the translation handle of the Polygon or Patch. The shape's geometry is not changed by doing this Ð it is only redrawn more closely about its local origin.

shapeScale;¬2.3.4 Scaling

A shape can be scaled in its local space using uniform or non-uniform scaling.

uniformScale;¬2.3.4.1 Uniform Scaling

In uniform shape scaling, the scale change is applied along all three axes equally and simultaneously. This is accomplished by first selecting the shape, mousedowning on the cyan scaling handle and dragging. Dragging to the right increases the scale; dragging to the left decreases it.

nonUniformScale;¬2.3.4.2 Non-Uniform Scaling

In non-uniform shape scaling, the scale change is applied only along one or two axes. This is accomplished by first selecting the shape, mousedowning on the cyan scaling handle with the Shift key down and dragging. Dragging horizontally left or right changes the scale along the x-axis. Dragging vertically up or down changes the scale along the y or z axes, depending on whether the Alternate key is up or down.

In addition, constraint can be applied to force scaling along one axis only.

shapeRot;¬2.3.5 Rotation

Shape rotation is accomplished by selecting the shape, mousedowning anywhere on the scene background and dragging. Rotation direction is determined by three factors:

- the direction the cursor is dragged in:
  - for rotation about the x-axis, drag vertically.
  - for rotation about the y-axis, drag horizontally.
  - for rotation about the z-axis, see the next two factors.

- the cursor's position relative to the rotator circle Ð see ;ThreeDmanual.rtfd;rotator;¬[1.1.3 Rotator Circle].
- the rotator axis control setting Ð see ;ThreeDmanual.rtfd;motionPanel;¬[1.2.2.1 Motion].

A secondary form of shape rotation also exists for approximating random shape positioning. See ;ThreeDmanual.rtfd;autoShapeRotation;¬[5.3 Auto Shape Rotation] for a description of this feature.

shapeGeom;¬2.3.6 Geometry

Each shape has its own geometry edit handles specific to its type. The following figures identify the geometry edit handles for each type of shape. Note that each type possesses a white translation handle and a cyan scaling handle in addition to the yellow geometry handles identified here:

835694_paste.eps ¬

The Teapot shape has no geometry edit handles.

geomHandles;¬2.3.7 Geometry Edit Handles

As noted above, certain shape types have geometry edit handles whose behaviour has been modified to reflect the peculiarities of the type's geometry. These modifications are as follows:

hyperboloid;¬2.3.7.1 Hyperboloid

The Hyperboloid edit handles Point1 and Point2 represent true, three-axis locations in space. They can be moved in the xy or xz planes with or without constraint in the same way that the translation handle is used.

patch;¬2.3.7.2 Patch

The Patch control point edit handles (one for each of the sixteen control points) represent true, three-axis locations in space. They can be moved in the xy or xz planes with or without constraint in the same way that the translation handle is used.

While moving any control point edit handle with the Shift key down, the positions of only the four corner control points are changed simultaneously.

polygon;¬2.3.7.3 Polygon

The Polygon vertex position handles (one per vertex) represent two-axis locations in space. They can be moved in the xy plane with or without constraint in the same way that the translation handle is used.

The number-of-vertices handle adds or removes vertices to the polygon. It can be single-clicked or dragged to accomplish this. Single-clicking adds a vertex for each click. Shift-single-clicking removes a vertex for each click. Dragging to the right adds vertices; dragging to the left removes them.

sphere;¬2.3.7.4 Sphere

The action of the Sphere radius edit handle can be modified using the Shift key.

When changing a sphere's radius with the Shift key up, the sphere always becomes a full sphere, even if it was truncated by Zmin or Zmax limits before the change.

When changing a sphere's radius with the Shift key down, the Zmin and Zmax limits are honoured and any sphere truncation is maintained during the radius change.

shapeColour;¬2.3.8 Colour

The colour of a shape can be changed at any time in the same way that the scene background colour is changed, as described in ;ThreeDmanual.rtfd;bgColour;¬[1.1.1.5 World Space Background Colour].

The shape is first selected, then the "Tools > Colors..." menu item is clicked to bring up the color panel. The desired colour Ð including opacity if required Ð is set , dragged from the panel and dropped anywhere into the scene view. The scene is then redisplayed with the shape having the new colour.

Note: The interactive scene window cannot display semi-transparent colours, even if opacities of less than 100 (more transparent) are set during colour specification. Such colours will appear fully opaque when displayed in the scene view.

However, semi-transparent colours will be correctly displayed if the scene is photoreal rendered, a procedure described in ;ThreeDmanual.rtfd;photorealRendering;¬[6 Photorealistic Rendering].

When creating shapes, ThreeD uses the same default colour for shapes of the same type. All cones are initially pink, cylinders initially red and so on.

shaders;¬2.3.9 Shaders

A **shader** is a function that is called by RenderMan during rendering to compute the appearance of each point on the shape. Shader functions are written in the RenderMan shader language and are stored in files with a `.slo` extension. A shader is set in ThreeD by specifying the file that contains it.

Warning: Specifying a shader file containing invalid or corrupt shader data will cause ThreeD to crash.

ThreeD performs no checks on the validity of the contents of any shader files it uses. It assumes that any `.slo` file is a valid RenderMan shader language function and attempts to use it as such.

ThreeD supports the use of all six standard RenderMan shader types. These types are:

- Surface
- Displacement
- Volume
- Transformation

´ Imager
´ Light

A shader is characterized by a number of parameters whose values can be changed to modify the way the shader works, thus modifying the appearance of the shape it helps to compute.

ThreeD allows the shaders of a shape to be changed and their parameters to be modified at any time. When ThreeD initially creates a shape it sets it to have the system-supplied surface shader called "plastic". This default surface shader can then be changed if desired. A newly created shape has none of the other shader types set for it Ð these can be manually specified by the user once the shape has been created..

Note that only the simpler shaders can be successfully used in the interactive scene. Most of the more complex (and therefore more interesting) shaders can only be properly used by rendering the scene photorealistically.

Shaders not usable in the interactive scene usually produce shapes having dark, muddy or granular appearances. This is not an error Ð it merely indicates that the particular shader chosen requires photoreal rendering to show its actual effect.

A frequently used shader combination uses a surface shader paired with a displacement shader. The surface shader supplies the basic appearance while the displacement shader provides the "bumpiness".

The Shape Inspector panel contains a splitview whose bottom half displays the current shader function and its parameter values for the selected shape if one is present. Use of this splitview is described in ;ThreeDmanual.rtfd;shaderMod;¬[2.3.9.3 Shader Parameter Modification].

shaderType;¬2.3.9.1 Shader Type

Directly below the shader information displayed in the Shape Inspector panel's splitview is a row of six radio buttons. These buttons serve as an indicator displaying the shader types for the selected shape. Each button corresponds to one of the six standard shader types: Surface, Displacement, Volume, Transformation, Imager and Light.

Shader type buttons that are enabled indicate the shader types that are currently defined for the selected shape. These are the shaders that will be used together when rendering the shape. A non-root shape will always have its surface shader defined.

A shader type's parameter information can be displayed at any time by clicking on the enabled button corresponding to that type.

Disabled buttons indicate the shader types that are not currently defined for the selected shape.

As long as a non-root shape remains selected, one of the buttons will always be highlighted. The highlighted button indicates which shader type's parameter values are currently being displayed in the splitview directly above. When a non-root shape is first selected its surface shader is the one that is initially displayed.

All buttons are disabled and unhighlighted when either no shape or the root shape is selected.

Setting a system or custom shader (described later) causes that shader's parameters to be displayed in the splitview and enables and highlights the button corresponding to the type of shader being set. ThreeD determines the type of shader being specified from the `.slo` shader file that defines the shader.

Note that displayed shader parameters are actually those of a particular shader *function* having the indicated shader type. When a shader is set or modified, it is this shader function which is being specified.

It is common to have many different functions that are all of the same shader type. For example, the "plastic", "wood" and "rmarble" shader functions are all classed as Surface type shaders and will be displayed as such by ThreeD. The "threads" and "diaknurl" shaders are Displacement types while "depthcue" and "fog" are Volume types, and so on.

systemShaders;¬2.3.9.2 Setting System Shaders

The Shape Inspector panel contains a scrollable list browser named "System Shaders" that displays the standard shaders stored on the system and available to ThreeD.

This list includes all shaders in the directory `/NextLibrary/Shaders`. ThreeD looks in this directory for files having a `.slo` extension and displays any it finds. If the list is empty it means that ThreeD could not find any shader files in this standard directory or was unable to access it for some reason.

A shape's shader is set from the system shader list by first selecting the shape. The list is next scrolled until the desired shader name is showing. The name is then selected by clicking on it, thus changing the shape's shader to the one selected. The shader display in the splitview will show the new shader and its current parameters,
and the shader type button corresponding to the type of shader specified will be enabled and highlighted.

customShaders;¬2.3.9.3 Setting Custom Shaders

In addition to the standard system shaders, ThreeD can accept and use custom shaders written in RenderMan shader language and compiled into `.slo` files.

A custom shader for a shape is specified by first selecting the shape as usual. The file icon of the `.slo` file containing the desired shader function is then dragged from the file viewer and dropped anywhere into the scene view. ThreeD will only accept files having the `.slo` extension. It ignores attempts to drop files of any other type.

As with system shaders, the shader display will show this new shader and its parameters, and enable and highlight the appropriate shader type button.

For a complete description of RenderMan shaders and how to write custom ones, see the book "The RenderMan Companion" by Steve Upstill, Addison-Wesley, 1990.

nonStdShaders;¬2.3.9.4 Setting Non-Standard Shaders

A shader may be of a type that is not one of the six standard shader types. Examples of such non-standard shaders include the "cylinmap", "automap" and "bilerp" shaders.

When given a non-standard shader to use, ThreeD cannot automatically determine how to apply it since the shader cannot be classified as belonging to any of the standard types recognized by the program. When this occurs, ThreeD will request that the user manually specify which standard type to use in order to apply the non-standard shader. This in effect forces ThreeD to treat the shader as belonging to one of the standard types even though it does not.

When a non-standard shader is encountered, ThreeD brings up a panel alerting the user to this fact. The panel contains a row of shader type buttons identical to the ones in the Shape Inspector panel. The button corresponding to the shader type desired for the non-standard shader is then clicked on. The parameter information of the shader is displayed in the splitview as usual and the shader type button indicating how the shader will be used is enabled and highlighted.

The same alert panel also contains a "Cancel" button which can be clicked on to cancel the operation, leaving the current shaders unchanged.

removingShaders;¬2.3.9.5 Removing Shaders

In addition to modifying the shader function of a particular shader type, the shader type itself can be removed from the shape. Doing this undefines that particular shader type for the shape, meaning that the shader type will not be applied to the shape when it is next rendered.

Removal of a shader from a shape is accomplished with the "Remove" button located to the right of the row of shader buttons.

A shader type is removed from a selected shape by first clicking on the shader type's button to display its parameter values in the splitview, then clicking on the "Remove" button. ThreeD unhighlights and disables the shader type button to indicate that the shader type is no longer defined for the shape. The Surface shader information is then redisplayed for the shape.

Note: The Surface shader type cannot be removed from a shape (although its shader function can be changed like any other type.) The Surface shader is the only required shader type for user-created shapes.

shaderMod;¬2.3.9.6 Shader Parameter Modification

A shader's parameters can be changed to effect a change in the shape's appearance.

This is accomplished by first selecting the shape, which causes its surface shader parameters to be displayed in the shader display splitview. If the surface shader is not the one desired, the button of the desired shader type can then be clicked on to display the required parameters. The parameter to be changed is scrolled into view if necessary and the line containing it is clicked on while holding down the Control key.

Shader parameters can be single numeric, multiple numeric, text or colour types. For each type, ThreeD presents a Shader Argument Inspector panel appropriate to the type. For example, changing a colour parameter brings up a panel containing a colour well as well as the standard colour panel. The desired colour is set and dragged from the panel into the well in the usual fashion.

Whatever its form, a Shader Argument Inspector panel always contains the name of the selected parameter and its current value. The new value is entered into the panel and the "set" button is clicked (or the Return key is pressed) to enter the new value.

If desired, the "Set Default" button can be pressed to obtain the RenderMan / 3DKit default value if one exists. (This is usually 1.0 and may not be what is desired.) Alternatively, the modification can be cancelled by clicking the "Cancel" button.

If a parameter is changed, its new value is shown in the shader display and takes effect the next time the shape is displayed as a solid surface.

grouping;¬2.3.10 Grouping

As described in ;ThreeDmanual.rtfd;hierarchy;¬[2.1.1 Shape Hierarchy], shapes are logically related to each other in a hierarchical structure called a shape tree. ThreeD allows a shape's position in this tree to be changed by a procedure known as **grouping**.

Grouping specifies that a shape will be made a descendant of some other shape, thereby giving it a new ancestor. This is accomplished by first selecting the shape that is to be the descendant, then holding down the Control key and clicking on the shape it is to be a descendant of.

For example, if "B" is to be a descendant of "A", "B" is first selected in the usual manner. "A" is then Control-selected. "B" is now a descendant of "A" and is said to be grouped to "A", its new direct ancestor.

When a shape is grouped in this way, it remains selected and its newly acquired ancestor is highlighted using wireframe. ThreeD always highlights all non-root ancestors of the selected shape in this way.

Changes made to the shape tree are immediately reflected in the shape hierarchy browser of the Shape Inspector panel. All non-root shapes highlighted in the browser will be highlighted using wireframe in the scene view.

A shape can be **degrouped** from its ancestor and returned to being a direct descendant of the root shape. This is done by selecting the shape to be degrouped, then selecting it again while holding down the Control key.

A shape cannot be grouped to one of its descendants. ThreeD will generate a warning if such a request is made and will not attempt the operation.

matchingPos;¬2.3.11 Matching Shape Positions

A shape in ThreeD can be made to match its position with that of another shape or with the world space axes. Location, orientation or both together can be matched, as described in the following sections.

locateToShape;¬2.3.11.1 Locate to Shape

A shape can made to translate itself such that its origin is coincident with that of another shape or with the world space origin. A shape thus relocated remains selected after the move and retains the same orientation it had prior to the move Ð it is only translated, not rotated.

Locating to another shape is accomplished by first selecting the shape to be moved, then holding down the Shift key and clicking on the shape to be located to. The selected shape will then move to place its origin on top of that of the other shape.

Locating a shape to the world space origin is accomplished by first selecting the shape to be moved, then holding down the Shift key and clicking on the same shape again. The selected shape will then move to place its origin on top of the world space origin.

orientWithShape;¬2.3.11.2 Orient with Shape

A shape can be made to rotate itself such that its axes are oriented with those of another shape or with the world space axes. A shape thus reoriented remains selected after the rotation and retains the same location it had prior to the rotation Ð it is only rotated, not translated.

Orienting with another shape is accomplished by first selecting the shape to be oriented, then holding down the Alternate key and clicking on the shape to be oriented with. The selected shape will then rotate to orient its axes with those of the other shape.

Orienting a shape with the world space axes is accomplished by first selecting the shape to be oriented, then holding down the Alternate key and clicking on the same shape again. The selected shape will then rotate to orient its axes with those of world space.

coincideWithShape;¬2.3.11.3 Coincide with Shape

A shape can be made to become fully coincident in space with another shape or with the world space axes. Coinciding with a shape is a combination of the above two operations Ð both the origins and the axes are made to coincide. A shape thus repositioned remains selected after the operation.

Coinciding with another shape is accomplished by first selecting the shape to be coincided, then holding down the Command key and clicking on the shape to be coincided with. The selected shape will then move and rotate to become coincident with the other shape.

Coinciding a shape with the world space axes is accomplished by first selecting the shape to be coincided, then holding down the Command key and clicking on the same shape again. The selected shape will then move and rotate to become coincident with the world space axes.

shapeSummary;¬2.4 Shape Operations Summary

The following table summmarizes the operations that can be performed on shapes:

Translation	**Rotation**	**Scaling**
Mousedown on white translation handle & drag. xy: Alternate xz: Alternate ↓	Mousedown on background & drag. Modify with: - rotation axis control - rotator circle	Mousedown on cyan scale handle & drag. Uniform: Shift Non-Uniform: Shift ↓ xy: Alternate xz: Alternate ↓
Locate to Shape	**Orient with Shape**	**Coincide with Shape**
P located to Q: - select P. - Shift ↓ & click on Q.	P oriented with Q: - select P. - Alternate ↓ & click on Q.	P coincident with Q: - select P. - Command ↓ & click on Q.

to World Origin:
- select P.
- Shift ↓ & click on P.

with World Axes:
- select P.
- Alternate ↓ & click on P.

with World Origin & Axes:
- select P.
- Command ↓ & click on P.

Grouping

P descendant of Q:
;ThreeDmanual.rtfd;polyPatchTrans;¬[2.3.3.1]
- select P.
;ThreeDmanual.rtfd;geomHandles;¬[2.3.7]
- Control ↓ & click on Q.

P descendant of root:
- select P.
- Control ↓ & click on P.

Single-Axis Constraint

Command ↓

Specific Edit Handles

Translation: see

Geometry: see

camera;¬3 Camera
ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

The camera is used to display the shapes within a scene. The following sections describe the operation of the camera in detail.

cameraBasics;¬3.1 Camera Basics

ThreeD provides a single camera whose location and orientation can be changed dynamically under user control. The two basic concepts of camera operation are the **scene window** and **standard positions**, both of which are described below.

sceneWindow;¬3.1.1 Scene Window

The scene window contains a single view that displays the scene as it appears when looking through the camera's lens and in effect acts as the camera's viewfinder. The scene view can be thought of as representing both the view through the camera and the camera itself. A full description of the contents of the scene window, including camera status and world coordinate display, is given in ;ThreeDmanual.rtfd;sceneWindowAndView;¬[1.1 Scene Window and View].

A scene can be viewed from a different position by simply moving the camera to the new position. This can be done either in one step by moving the camera to one of its standard positions, described in the next section, or in a dynamic, continuous fashion using the mouse. All mouse-directed camera motion is performed within the scene view as described in ;ThreeDmanual.rtfd;cameraMotion;¬[3.2 Camera Motion].

stdPositions;¬3.1.2 Standard Positions

In addition to being dynamically movable through a scene, the camera can be moved in a single step to one of seven predefined positions, each of which is called a **standard position**. These positions make it possible to quickly move the camera to a known position in world space, a capability which can be useful if extensive camera motion within a complex scene has made its orientation with respect to scene shapes uncertain.

Each standard position locates the camera on a specific world space axis at a preset standard position distance from the world space origin, orienting the camera to point directly at that origin. The camera's local coordinate space is always oriented such that its x-axis runs horizontally across the scene window, its y-axis runs vertically up and down the scene window and its z-axis runs into and out of the scene window.

The seven standard positions and their associated world space axes are:

- Front   on negative Z-axis
- Back   on positive Z-axis

- Left			on negative X-axis
- Right			on positive X-axis
- Top			on positive Y-axis
- Bottom		on negative Y-axis
- On Origin		on world space origin, same orientation as Front

Below is a diagram showing where each standard position lies in world space along with the world space axes. Each position is shown as a labelled, solid white circle with an attached arrow indicating the direction the camera points in when at that position.

paste.tiff ¬

Standard Positions in World Space

When ThreeD is started the camera is initially set at its default standard position of Front at a standard position distance of 5 units from the world space origin.

stdPosSelection;¬3.1.2.1 Standard Position Selection

The camera can be moved to a standard position by first selecting the Setup panel's "Camera" subpanel. This subpanel displays the Standard Positions control group containing the Positions pulldown list. This list is activated and the desired standard position is selected from it.

Alternatively, a standard position can be selected by entering the Command key combination corresponding to it. These are shown on the Positions pulldown list in the usual manner. Command-1 selects Front, Command-2 selects Back and so on.

Selecting a standard position causes the camera to move immediately to that new position using the preset standard position distance currently in effect.

stdPosDistance;¬3.1.2.2 Changing Standard Position Distance

The standard position distance from the world space origin can be changed using the text field   named "Distance Away", located next to the Positions pulldown list.

To change the standard position distance the current value in the "Distance Away" text field is selected in the usual fashion, the new distance value is entered and the Return key is pressed. ThreeD accepts any non-zero, positive real number as a distance value.

Changing the standard position distance causes the camera to move immediately to the most recently selected standard position and use the new distance. The new distance value becomes the current one and is applied to all standard positions that are subsequently selected. It remains in effect until explicitly changed.

Note: A standard position with a distance of zero can be obtained by selecting the On Origin standard position since the latter is coincident with the world space origin.

stdPosAnnotation;¬3.1.2.3   Standard Position and Annotation

Whenever the camera is moved to a new standard position or distance, the result is immediately shown in the scene view and the scene annotation described in ;ThreeDmanual.rtfd;annotation;¬[1.1.2 Annotation] is updated to reflect the new position. The camera status line along the top will indicate the new standard position and distance of the camera while the world coordinates of the camera at that standard position will be displayed along the bottom.

Any camera movement away from a standard position causes ThreeD to display the message "Camera in non-standard position." in the camera status line. Even very small deviations in camera location and/or orientation from a standard position will cause this status message to be displayed.

cameraMotion;¬3.2 Camera Motion

The camera can be translated, rotated and flown as described in the following sections.

cameraConv;¬3.2.1 Conventions

All camera motion occurs within the camera's local coordinate space, relative to the orientation of its local coordinate axes.

For all linear motions along the camera's axes, the following conventions are adhered to. These are identical to those used for shape changes:

- x-axis change:
  - towards positive end of axis   move cursor horizontally to right
  - towards negative end of axis   move cursor horizontally to left

- y-axis change:
  - towards positive end of axis   move cursor vertically upwards
  - towards negative end of axis   move cursor vertically downwards

- z-axis change   same as y-axis but with Alternate key
down

- constrain change to one axis move cursor with Command key down

The constraint feature is discussed in ;ThreeDmanual.rtfd;changeConstraint;¬[2.3.1.1 Change Constraint]. As usual, ThreeD deems the constrained axis to be the one associated with the larger initial displacement when the mouse starts to move with the Command key held down.

By following these conventions, linear camera motions can be applied in either the xy or the xz plane along both axes simultaneously or constrained to only one of them.

Like all dragging operations, linear camera motion is affected by the drag sensitivity setting as described in ;ThreeDmanual.rtfd;motionPanel;¬[1.2.2.1 Motion].

cameraTrans;¬3.2.2 Translation

The camera is translated by double-click mousedowning anywhere in the scene background and dragging in the xy or xz planes, constraining motion to one axis if desired.

During translation, the camera's world X, Y and Z coordinates are displayed in the bottom annotation line as they change. The particular type of camera translation being performed is displayed in the top annotation line at the same time.

cameraRot;¬3.2.3 Rotation

The camera is rotated by double-click mousedowning anywhere in the scene background while holding down the Shift key and dragging. Rotation direction is determined by the same three factors that determine shape rotation:

- the direction the cursor is dragged in:
    - for rotation about the x-axis, drag vertically.

- for rotation about the y-axis, drag horizontally.
- for rotation about the z-axis, see the next two factors.

· the cursor's position relative to the rotator circle Ð see ;ThreeDmanual.rtfd;rotator;¬[1.1.3 Rotator Circle].
· the rotator axis control setting Ð see ;ThreeDmanual.rtfd;motionPanel;¬[1.2.2.1 Motion].

During rotation, the camera's static world X, Y and Z coordinates are displayed in the bottom annotation line. Since rotation is about the camera's origin and does not alter its location in world space, the coordinates do not change. The particular type of camera rotation being performed is displayed in the top annotation line at the same time.

flight;¬3.2.4 Flight

The camera can be flown through world space by triple-click mousedowning anywhere in the scene background. Flight consists of camera rotation combined with translation along either its z-axis or one of the world axes.

While flying, the camera can be rotated with the same mouse dragging action used for in-place rotations. Rotation will be according to the rotator axis setting and the rotator circle as usual.

The Interaction panel's Motion subpanel contains a control group named "Flight" which further contains two control subgroups, one named "Direction" and the other "Speed" for controlling camera flight.

flightDirection;¬3.2.4.1 Flight Direction

Flight direction can be set to be in the direction the camera is pointing at any given instant or parallel to one of the world axes. These directions are set by clicking one of the Direction radio buttons.

When flight direction is determined by camera pointing direction, rotating the camera during flight by dragging the mouse changes the camera's pointing direction, thereby changing flight direction, allowing in-flight course changes to be made.

Alternatively, flying parallel to one of the world axes gives an impression of performing a "fly-by" when moving past shapes. The camera is in effect sitting on a platform which is flying along a line parallel to the chosen world axis. The camera itself is free to rotate in any direction while riding on the platform and can be swivelled about to keep shapes in view as they are passed.

flightSpeed;¬3.2.4.2 Flight Speed

Flight speed is set using the Speed slider, from slow to fast.

Flight speed can be also be set to positive or negative by clicking one of the Speed radio buttons. If flight direction is Where Pointed, Positive flight is forward and Negative is backward. If flight direction is parallel to one of the world axes, Positive is in the positive direction of the axis and negative is in the axis' negative direction.

lookAt;¬3.2.5 Look At

The camera can be made to point directly at the currently selected shape (if one exists) or at the world space origin.

This "look at" capability is useful in cases where the selected shape (or the world space origin) is not in view and it isn't known in what direction the camera should be pointed to see it again. This can arise when shapes become "lost", a situation which is described in ;ThreeDmanual.rtfd;lostShapes;¬[2.2.1.3 Lost Shapes].

The Interaction panel's Motion subpanel contains a control group named "Look At" which has two buttons. If a selected shape exists, clicking the button named "Selected Shape" causes the camera to point at it. This button is disabled when no shape is selected. Clicking the button named "World Origin" always points the camera at the world space origin.

"Look At" works by rotating the camera about its local origin until it points at the selected shape or the world origin. Because the camera is only rotated and never translated during this process, its location in world space remains unchanged Ð only its orientation is affected.

Once this has been accomplished, the selected shape's local origin (as marked by the shape's white translation handle) or the world origin will be visually centered within the scene view.

Note: "Look At" moves the camera somewhat arbitrarily through whatever rotations are required to accomplish centering. It does not attempt to preserve any camera orientation that existed prior to its action.

As a result, the final orientation of the camera relative to the centered shape (or world origin) may differ significantly from what it was when the object was last in view. However, the object will always be centered within the scene view and the camera's location in world space will always remain unchanged.

cameraSummary;¬3.3 Camera Operations Summary

The following table summmarizes the operations that can be performed on the camera:

Translation	Rotation	Flying	
Double-click mousedown on background & drag. xy: Alternate xz: Alternate ↓ - rotator circle	Shift ↓ & double-click mousedown on Modify with:	Triple-click mousedown on background & drag. Modify with Flight controls.	background & - rotation axis

Single-Axis Constraint

Command ↓

lights;¬4 Lights

ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

ThreeD supplies three light sources with which to light a scene: an **ambient** light, a **distant** light and a **spot** light. All of these lights are global in the sense that all shapes in a scene are illuminated by them.

These three lights constitute ThreeD's sole lighting capability. They cannot be removed and no additional lights can be created.

The following sections describe ThreeD lighting.

lightIntensity;¬4.1 Light Intensity

The intensity of each light can be varied independently from low to high.

The Setup panel's Lights subpanel contains a control group named "Intensity" which has three vertical sliders, each labelled according to the light whose intensity it controls. The slider for the desired light is moved up and down to vary the light's intensity. The bottom position marked "Lo" represents a zero level intensity and can be used to turn a light off completely.

As an intensity slider is moved, the resulting change in scene illumination is dynamically displayed in the scene view.

ambientLight;¬4.2 Ambient Light

All shapes in a scene use the ambient light for self-illumination. Unlike the other two lights, the ambient light is non-directional and has no specific location in space, existing instead as if it covered the surface of each shape. Even when the other two lights are turned off, a shape can still be illuminated at the scene's ambient light level, which is used to supply each shape with a kind of "glow from within".

distantLight;¬4.3 Distant Light

The distant light is directional. It is located in world space at coordinates X = -30, Y = -30, Z = -30 and points directly at the world origin. This position is fixed and cannot be changed.

When the camera is set at standard position Front at a distance of 5 units from the world origin, the distant light is behind and to the lower-left of the camera.

The distant light's intensity can be changed as described above in ;ThreeDmanual.rtfd;lightIntensity;¬[4.1 Light Intensity]. Its illumination does not fall off with distance.

spotLight;¬4.4 Spot Light

The spot light is also directional and is located in world space at coordinates X = 0, Y = +4, Z = 0. This location is fixed and cannot be changed Ð it is anchored to the positive Y-axis. When creating a new scene, ThreeD sets the spot light by default to point directly at the world origin.

When the camera is set at standard position Front at a distance of 5 units from the world origin, the spot light is located above and in front of the camera.

The spot light's intensity can be changed as described above in ;ThreeDmanual.rtfd;lightIntensity;¬[4.1 Light Intensity]. Its illumination falls off with distance.

xzAngles;¬4.4.1 X and Z Angles

Although the location of the spot light is fixed, the direction it points in can be changed. This is accomplished by rotating it through an angle that sweeps its beam along a line parallel to the world space X-axis and/or the world space Z-axis. By combining both motions the spot light can be pointed to any location on the world space XZ plane. This is illustrated in the diagram in ;ThreeDmanual.rtfd;coneAngle;¬[4.4.2 Cone Angle].

The Setup panel's Lights subpanel contains a control group named "SpotLight Angles" which has three horizontal sliders. The upper two are labelled "X" and "Z", indicating which world space axis the spot light's beam will be swept parallel to. Each can be independently controlled.

The left end of each slider is marked with a negative sign and the right end is marked with a positive sign. These indicate in which direction on the world space axis the beam will be swept along. Moving a slider towards its positive end moves the beam towards the positive end of its corresponding world space axis, while the opposite holds for negative directions.

As the spot light's angle sliders are moved, the resulting change in scene illumination is dynamically displayed in the scene view.

coneAngle;¬4.4.2 Cone Angle

The spot light's illumination cone angle can be changed, thereby altering the diameter of the spot light's illumination circle.

The lowest of the three horizontal sliders in the"SpotLight Angles" control group described above is labelled "Cone". Moving this slider changes the spot light's cone angle from "Narrow" at the far left to "Wide" at the far right.

As the cone angle slider is moved, the resulting change in scene illumination is dynamically displayed in the scene view.

Below is a diagram showing the spot light's location in world space along with the world space axes. Shown in white are two possible illumination cones pointing down to illuminate different locations on the world XZ plane shown in blue.

482283_paste.tiff ¬

Spot Light Illumination Cones

sequencing;¬5 Sequencing

ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

ThreeD possesses a limited ability to generate a scene **sequence** and to display this sequence repetitively.

A sequence consists of two or more renderings of the same basic scene. Each of these separate renderings is called a **frame**. For each successive frame, small incremental changes are made to the positions of shapes within the scene. Thus, each frame in a sequence represents a unique configuration of shapes in space, with each frame showing the shapes in positions slightly different from those in the frames before and after it.

If frames are generated in this way and shown one after the other in succession, the resulting display will be of a scene whose shapes appear to be moving in space, thereby achieving a degree of scene **animation**.

Note: ThreeD's sequencing capability is distinctly limited and offers very little user control over the final result Ð animations obtained with ThreeD are rudimentary at best. However, the application does demonstrate basic principles of scene animation.

The following sections describe sequencing in detail.

seqGeneration;¬5.1 Sequence Generation

ThreeD generates a sequence by first applying a constant incremental rotation to every shape in the scene, including the root shape, and then rendering the scene as a frame. All shapes receive the same amount of incremental rotation.

Subsequent frames in the sequence are created by repeating this process Ð the incremental rotation is applied to every shape once more and the scene is re-rendered. As frames are generated the incremental rotations accumulate, effecting a continuous change in the orientation of each shape within the scene.

Sequence generation in ThreeD involves three things: the rotation axis, the rotation increment and the number of frames. These are described next.

seqRotationAxis;¬5.1.1 Rotation Axis

Like all motion in ThreeD, the incremental rotation is applied within each shape's local coordinate space, relative to the orientation of its local coordinate axes. The local axis about which a shape is rotated during frame generation is arbitrarily set by ThreeD and cannot be changed Ð it is "hard-wired" into the program and remains constant.

A different rotation axis is assigned for each different shape type. Cones have their own rotation axis, cylinders have theirs and so on. All user-created shape types have rotation axes lying between their principal x, y and z axes. The root shape is always rotated about its y-axis.

Because all user-created shapes are descendants of and are grouped to the root shape, they will all rotate along with the root shape when it is rotated during sequence generation. As a result, each shape's effective incremental rotation will be a combination of its own local rotation and that of the root shape overlaid on top of it.

Note: Although a shape's sequence rotation axis is set by ThreeD and cannot be changed, the appearance of its sequence rotation in world space can be modified.

This is accomplished by rotating and/or translating the shape manually to a new position before generating the sequence. The result is to change the position of the shape's local coordinate axes relative to those of the root shape. When the user-created shape rotates along with the root shape its new position relative to the latter gives its sequence rotation a new appearance.

seqRotationInc;¬5.1.2 Rotation Increment

The rotation increment Ð the amount by which each shape is rotated per frame Ð cannot be directly specified. Instead, the increment is a function of the number of frames in the sequence, a parameter which can be changed. Setting the number of sequence frames is described in the next section.

ThreeD always sets the increment to a value such that when all frames in the sequence have been generated and displayed, each shape will have been rotated by 360°. The increment is inversely proportional to the number of frames. If there are 10 frames in the sequence, the increment is set to 36° per frame, 20 frames uses 18° per frame, and so on.

numSeqFrames;¬5.1.3 Number of Frames

The Interaction panel's Sequencing subpanel contains a horizontal slider named "# Frames to Generate". The number of sequence frames currently in effect is always displayed in the text field to the right of the slider.

The number of sequence frames to be generated can be varied from a minimum of 1 to a maximum of 100 by moving the slider from left to right. When ThreeD starts up, the default value of 1 is always set.

Note: For consistency, ThreeD treats normal, single renderings of a scene as one-frame sequences.

seqPlayback;¬5.2 Sequence Playback

The sequential display of frames is called **playback**. A frame sequence generated by ThreeD in the above manner can be played back for viewing in the interactive scene window.

The Interaction panel's Sequencing subpanel contains a button named "Play Once Interactively". Clicking this button causes the frames of the current scene to be generated and displayed one after the other in the scene window.

The sequence is played back once each time the button is clicked. The number of frames generated is determined by value of the "# Frames to Generate" slider in effect when the button is clicked. Frames are regenerated anew for each playback Ð thay are not generated once and then stored for subsequent repeat playbacks.

autoShapeRotation;¬5.3 Auto Shape Rotation

During normal scene creation, each shape is individually translated and/or rotated on its own in order to place it in a specific position in space. However, there may be times when a "random" positioning of all the shapes in a scene is desired.

Such positioning can be approximated by selectively turning on and off the process ThreeD normally uses to automatically rotate shapes during sequence generation.

The Interaction panel's Options subpanel contains a control group named "Shapes During Interaction". This group has a checkbox named "All Shapes Auto-Rotate" whose normal default state is unchecked Ð ThreeD's process of automatic shape rotation is turned off and manual shape positioning is done in the usual fashion.

However, whenever this checkbox is checked, automatic shape rotation is turned on. ThreeD will automatically apply the same rotations it uses to generate a sequence to every shape in the scene, including the root shape.

Note: The rotation increment used is determined by the setting of the "# Frames to Generate" slider. This slider must be set to a value greater than 1 for automatic shape rotation to be noticeable.

This automatic application of shape rotation is done while the selected shape is being manually rotated or translated. The auto rotation is added to the manually applied movement of the selected shape while all other shapes, including the root shape, are simultaneously auto-rotated.

The visual effect is to cause a simultaneous, apparently random re-positioning of all shapes in the scene.

photorealRendering;¬6 Photorealistic Rendering

ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

ThreeD can render a scene photorealistically. The following sections describe this process.

renderAccuracy;¬6.1 Rendering Accuracy

Two levels of accuracy are available when rendering a scene in ThreeD. These levels differ in the amount of scene detail they generate and in the time required to compute it. The following sections describe the two levels.

QRMAccuracy;¬6.1.1 Quick RenderMan Accuracy

During interactive operation, a ThreeD scene is created, manipulated and displayed in the interactive scene window. The scene's visual appearance in this window is obtained by ThreeD through a rendering operation performed by the Quick RenderMan program, which computes the scene's appearance as the latter is created and modified interactively.

Computation of a scene in this manner requires that the rendering program be fast enough to be usable under interactive conditions. To be sufficiently responsive, Quick RenderMan computes only an approximation of the scene; an approximation which lacks detail and fidelity compared to the ideal "original". Visual accuracy is traded off to gain rendering speed when operating in an interactive environment.

PRMAccuracy;¬6.1.2 Photoreal RenderMan Accuracy

In addition to the faster, lower accuracy rendering of Quick RenderMan, ThreeD can generate a more accurate rendering of the scene through a process known as **photoreal rendering**.

This form of rendering attempts to generate a scene with "photographic" fidelity. The shapes in a photoreal scene are more accurate and detailed in their surface geometry and appearance compared to those rendered interactively by Quick RenderMan. ThreeD uses the Photoreal RenderMan program to perform photoreal rendering.

However, it takes much longer to compute a scene photorealistically than it does to compute the same scene with Quick RenderMan Ð rendering speed is traded off to gain visual accuracy. Photoreal rendering can take minutes or even hours rather than seconds.

Because of the lengthy processing times involved, the Photoreal RenderMan program runs as an independent process separate from ThreeD. This process does the work of computing the photoreal scene, freeing ThreeD for continued interactive use while the computation proceeds. When the photoreal process has computed all frames in the sequence, it notifies ThreeD which then displays the photoreal results.

renderingTimes;¬6.2 Rendering Times

Photoreal rendering time is a function of image size, resolution, number of shapes, and shape and shader complexity. For example, a simple scene consisting of a default-sized cone and cylinder with plastic shaders in a scene window of default size can require a photoreal rendering time of 80 seconds on a lightly loaded 25Mhz M68040 NeXTSTATION Colour system with 16MB of memory.

A 50 frame sequence on the same system would therefore take a little more than one hour to render. More complex scenes, especially larger ones having more sophisticated shaders can take considerably longer than this. The photoreal rendering of multi-frame sequences is discussed in greater detail in the next section.

multiFrameRendering;¬6.2.1 Multi-Frame Sequence Rendering

The photoreal process always renders the entire number of scene frames set by the "# Frames to Generate" slider described in ;ThreeDmanual.rtfd;numSeqFrames;¬[5.1.3 Number of Frames].

This may not always be what is desired. Multi-frame sequences are usually viewed first with Quick RenderMan in their entirety since rendering times at this accuracy level are short enough to be acceptable.

However, because of the much longer processing times required for photoreal rendering, often only the first frame of a sequence is test-rendered at a photoreal level before commiting to a full-sequence photoreal rendering session.

First-frame photoreal rendering is done simply by setting the frame slider back to 1 and then initiating the rendering Ð only the first frame will be photoreal rendered. Once rendering starts, the number of frames can be immediately increased again if desired.

If photoreal rendering is inadvertently started when a sequence is set to have more than one frame, a significant amount of unnecessary computation will be performed. To prevent this from happening, ThreeD always issues a warning in the form of an alert panel indicating that multiple frames will be rendered. At this point rendering can either be cancelled or okayed by clicking the appropriate button on the panel.

This warning is given for sequences having two or more frames.

performingRendering;¬6.3 Performing Rendering

Any scene in the interactive scene window can be photorealistically rendered. Photoreal rendering is initiated using the Render panel and the results are viewed in the image window. These components are described in the following sections.

renderPanel;¬6.3.1 Render Panel

The Interaction panel's Options subpanel contains a button named "Render PhotoReal". Clicking on this button brings up a second window titled "PhotoReal Image" and a panel titled "Render". The image window is described in ;ThreeDmanual.rtfd;imageWindow;¬[6.3.2 Image Window].

The Render panel is used to specify the image resolution of the photoreal result and the name of the computer that should perform the photoreal rendering computations.

The desired image resolution in dots per inch (dpi) can be entered in a small text field for that purpose. The default resolution of 72dpi will already be set Ð this is the default value used when displaying a photoreal scene on the screen.

The rendering computer name(s) are displayed in a scrollable list, and the name of the computer to do the rendering is clicked on to select it.   For systems on a network, more than one computer name may be available. For a standalone system, of course, the only name displayed will be that of the standalone system, and it will already be selected by default.. In most cases this name will be "localhost".

Once the resolution and rendering computer have been set or defaulted, the "OK" button is clicked and the photoreal rendering process is started. Alternatively, the "Cancel" button can be clicked at this point to cancel photoreal rendering.

imageWindow;¬6.3.2 Image Window

The image window appears automatically when photoreal rendering is initiated for the first time in a ThreeD session. Its default title is "PhotoReal Image".

This window can be miniaturized and closed in the usual fashion but cannot be user-resized. Closing it does NOT delete the photoreal sequence displayed within it Ð it merely removes the window from the screen. A closed image window can be re-opened at any time by clicking the "Tools > Image Window" menu item.

Although not user-resizeable, the image window does automatically resize itself to the dimensions of the photoreal image it displays. If the interactive scene window is made larger or smaller, a subsequent photoreal rendering of the resized scene will cause the image window to resize itself correspondingly before displaying the scene.

This resizing can also occur when reading TIFF image files of previously photoreal rendered scenes, a process described in ;ThreeDmanual.rtfd;readingPhotoreal;¬[8.2 Reading in Photoreal Rendered Frames].

Once rendering is initiated from the Render panel (see ;ThreeDmanual.rtfd;renderPanel;¬[6.3.1 Render Panel]) the title of the image window is changed to "Rendering in progress..." while the computations are being performed.

When they are complete, ThreeD sounds a beep, changes the image window title to "Rendering completed" and displays the photoreal scene in it. A control panel titled "PhotoReal Playback" is also displayed to the right of the image window. Its use is described in ;ThreeDmanual.rtfd;playbackControl;¬[6.5 Playback Control].

If a multi-frame sequence has been rendered, the first frame in the sequence is the one initially displayed.

monTermRendering;¬6.4 Monitoring and Terminating Rendering

The separate photoreal rendering process can be monitored and terminated as described in the following sections.

Note: The following discussion is based on RenderMan behaviour as bundled with Version 3.0 of NeXTSTEP, running on a single, standalone system. This behaviour may be different on systems running later versions of NeXTSTEP or on systems residing on a network.

monRendering;¬6.4.1 Render Monitoring

The process that is rendering the current frame can be monitored by issuing the shell command "`ps -x`" from a terminal window. This command displays a list of processes running on the system which will include the photoreal rendering process shown on a line similar in form to the following:

```
PID TT STAT  TIME COMMAND
.
.
213 ?  R     0:32 prman tiffrender_5.000.rib
.
.
```

The above example indicates that a process with a PID of 213 has invoked the

Photoreal RenderMan command "`prman`" to render a frame from its RIB code description in temporary `.rib` file "`tiffrender_5.000.rib`"and will store the rendered result in the form of a TIFF image for subsequent display by ThreeD. The process is in state "`R`" (runnable) and has currently used 32 seconds of CPU time.

When a single system is used to perform photoreal rendering, the frames in a sequence are rendered one at a time in reverse sequence order. At any given instant there will be one process rendering one frame on the system, starting with the highest frame number and working down.

As each frame is completed, the process that computed it is deleted and a new one with a new PID is started to compute the next frame. The name of the temporary `.rib` file displayed in the process listing indicates which frame is currently being worked on. This filename has the form:

`tiffrender_`*`integer`*`.000.rib`

where `integer` represents the number of frames still to be rendered. The last frame to be rendered is numbered one, not zero.

termRendering;¬6.4.2 Render Terminating

A photoreal rendering operation can be prematurely terminated by the user on either a current frame basis or on an all-frames basis. The procedures for doing this are described next.

termCurrentFrame;¬6.4.2.1 Terminating the Current Frame

The process rendering the current frame can be terminated by issuing the shell command "`kill` *`pid`*" from a terminal window, where *`pid`* is the PID of the photoreal process to terminate. This PID can be obtained by issuing the "`ps -x`" command described above.

When a process rendering a non-final frame in a multi-frame sequence is terminated in this way, the system will immediately start a new process to begin rendering the next frame.

termAllFrames;¬6.4.2.2 Terminating All Frames

To terminate processing of *all* remaining frames at once, thereby aborting the entire rendering operation, it is necessary to terminate a background process known as the **rendering daemon**. Doing this requires superuser privileges and can be done by logging on as the "root" user.

CAUTION: Terminating processes with superuser privileges or when logged on as "root" can be dangerous to your system's integrity if not done correctly.

The following discussion is directed to users who are familiar with UNIX system administration. If you are not familiar with UNIX processes at a system management level, you should probably NOT attempt the following procedure. Rendering processes can always be stopped by simply rebooting your system according to the NeXTSTEP manual Ð not an elegant method but one that is less risky than terminating specific system processes if you're not sure of what you're doing.

Once logged on as "root", the shell command "`ps -ax`" is first issued from a terminal window to obtain the PID of the rendering daemon. The resulting process list will include the rendering daemon process, which is identifiable by having the entry "`rpc.renderd`" in the COMMAND column of the listing.

To terminate this process, the command "`kill` *`pid`*" is issued, where *`pid`* is the PID of the rendering daemon process obtained from the process listing just described.

If a photoreal process happens to be computing a frame when the render daemon is terminated, the photoreal process runs to completion and then terminates automatically. No new processes are started for any subsequent frames, since starting them is the job of the daemon process which no longer exists.

The render daemon process will be automatically restarted the next time photoreal rendering is initiated.

tempRenderFiles;¬6.4.2.3 Temporary Rendering Files

If the rendering daemon is terminated as described above, a number of undeleted, temporary `.rib` and `.tiff` files may remain behind since the process was unable to perform normal file cleanup due to the abrupt nature of its termination. Such files may also remain if ThreeD itself is Quit before photoreal rendering is completed.

These files reside in a temporary directory created and owned by "root". All of the files and the directory containing them can be deleted manually as the "root" user. This is usually desireable since they often consume significant amounts of disk space.

Note: As a general rule, care should always be exercised whenever deleting files as the "root" user.

The temporary directory has a name of the form:

`/usr/spool/render/job.`*`integer`*

where *`integer`* is a rendering job number identifying the particular photoreal rendering session that created the temporary files and directory. There is usually a `.rib` file for each frame in the sequence and each `.rib` file may have a corresponding `.tiff` file, depending on how far rendering had progressed when terminated.

playbackControl;¬6.5 Playback Control

When ThreeD finishes photoreal rendering a scene, it displays the first frame of the sequence in the image window. If the sequence consists of only one frame, the photoreal operation is complete Ð the single frame displayed is the final result of the rendering procedure.

However, if a multi-frame sequence has been rendered, ThreeD has the ability to display each of the frames in the sequence one after the other in the image window in an operation called **playback**.

Sequence playback is performed using the controls contained in a panel titled "PhotoReal Playback". This panel is automatically displayed next to the image window whenever photoreal rendering has been completed.

The playback panel is used to control auto and manual playback as well as sequence deletion. These operations are described in the following sections.

autoPlayback;¬6.5.1 Auto Playback

The playback panel contains a control group named "Auto" which has a "Start" and a "Stop" button. Clicking the "Start" button causes ThreeD to begin displaying each frame one after the other in the image window. When all frames in the sequence have been displayed once, playback immediately begins again at the first frame, displaying the sequence over and over as an endless loop.

Sequence playback continues in this manner until the "Stop" button is clicked. ThreeD always plays a sequence in its entirety before checking to see if it should stop auto playback. As a result, there is usually a delay between clicking the "Stop" button and the actual termination of playback while ThreeD plays out the sequence to its final frame.

playbackSpeed;¬6.5.1.1 Playback Speed

Playback speed cannot be controlled Ð ThreeD plays frames back as quickly as it can. When a sequence is played back for the first time, there will often be noticeable delays between frames as the image data is read in from disk. This creates a visually jerky appearance during initial playback.

However, once all frames have been displayed once and all image data is resident in memory, these delays are reduced and playback usually becomes faster and smoother.

Playback speed is primarily a function of system load and memory availability. Playback tends to be halting and slow on heavily loaded systems or on systems having insufficient amounts of memory.

manualPlayback;¬6.5.2 Manual Playback

The playback panel contains a control group named "Manual" which has a horizontal slider and a text field labelled "Frame #". The slider can be moved back and forth to manually display individual frames in the sequence. Frame display is continuously updated while the slider is moved, with the number of the currently displayed frame being likewise updated and displayed in the text field.

Moving the slider to the far left position labelled "first" displays the first frame, while moving it to the far right position labelled "last" displays the last frame. Moving through intermediate positions causes intermediate frames to be displayed.

Forward and reverse playback can be performed simply by moving the slider to the right and left.

seqDeletion;¬6.5.3 Sequence Deletion

The playback panel contains a button labelled "Delete Image Sequence". Clicking this button deletes the entire image sequence from memory, clears the image window and resets its title to "PhotoReal Image".

Deleting an unwanted sequence in this way can be beneficial since it frees up what is often a significant amount of memory, making it available to other applications including ThreeD. Such manual deletion is not mandatory, however, since ThreeD automatically deletes any previously rendered sequence from memory before commencing to render a new one.

ThreeD issues an alert if an attempt is made to delete a photoreal sequence which has never been saved to disk. At this point deletion can be cancelled or proceeded with after optionally saving the sequence first.

Note: A sequence can be obtained by reading in previously saved image files. Deleting a sequence in the above manner does NOT delete these image files or affect them in any way. The sequence is deleted from memory only and the files remain untouched.

savingScenes;¬7 Saving Scene Data

ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

ThreeD interactive scenes and photoreal rendered frames can be saved to files and read back in later for further scene modification and frame display. They can also be copied to the pasteboard. The following sections describe saving and copying scene data in detail.

savingInteractive;¬7.1 Saving Interactive Scenes

A scene being displayed in the interactive scene window can be saved to a file in two possible representations Ð as a **scene file** and as a **RIB file**. An interactive scene can be saved at any time in either or both of these forms.

The following sections describe scene saving using the two possible representations.

sceneFiles;¬7.1.1 Scene Files

A scene file is the usual form in which to save an interactive scene. Scenes must be saved in this form in order to be read in later for further modification.

Note: To save an interactive scene to a scene file, the interactive scene window must first be made the key window.

A scene can be saved to an explicitly named scene file by clicking on the "File > Save As..." menu item or by entering its keyboard equivalent of Command-S. Doing this brings up a standard save panel titled "Save Scene (.threeD)". The desired directory is specified and a scene filename is entered in the text field labelled "Scene Name:".

Clicking the "OK" button saves the scene to a file with the specified name. ThreeD always appends a file extension of `.threeD` to the filename Ð it is not necessary to specify an extension when entering the name. This `.threeD` extension is used by ThreeD to denote any scene files it creates.

ThreeD scene files are displayed using the following document icon:

ThreeDDoc.tiff ¬

.threeD Document Icon

A scene can also be saved to an implicitly named scene file by clicking on the "File > Save" menu item or by entering its keyboard equivalent of Command-s. Doing this saves the scene directly to the file that the scene was last explicitly saved to or read in from, without bringing up the save panel. If a scene file has never been explicitly named, ThreeD prompts for a name by bringing up the save panel.

If the current scene in the interactive scene window has not been saved, any operation that will delete the unsaved scene data causes ThreeD to first issue an alert, warning that the scene has been edited (but not saved). At this point the operation can be cancelled or proceeded with after optionally saving the scene first.

RIBFiles;¬7.1.2 RIB Files

"RIB" is an acronym for "RenderMan Interface Bytestream", a scene description language used to specify the appearance of three-dimensional scenes. It is analogous to Postscript, a language for specifying the appearance of two-dimensional pages.

A RIB file contains RIB language code describing the visual characteristics a particular three-dimensional scene. The code itself is in the form of English-like text consisting of keywords and data which can be examined with a normal text editor.

The RIB code describing a scene is usually generated by a scene modeling program. This code can then be passed to a rendering program which interprets the RIB code, performing computations based on that code to render the scene as a visible image.

A RIB description contains no information about the visual accuracy or resolution of a scene's appearance Ð this is determined by the rendering program that uses the RIB description.

A ThreeD scene's RIB code can be saved directly to a file for subsequent use by other programs, such as renderers, that understand RIB code.

Note: Unlike saving to a scene file, saving to a RIB file does *not* require that the interactive scene window first be made key.

A scene's RIB code can saved to an explicitly named file by clicking on the "File > Save RIB As..." menu item. Doing this brings up a standard save panel titled "Save RIB (.rib)". The desired directory is specified and a filename for the RIB code is entered in the text field labelled "RIB Name:".

Clicking the "OK" button saves the scene's RIB code to a file with the specified name. ThreeD always appends a file extension of `.rib` to the filename Ð it is not necessary to specify an extension when entering the name. This `.rib` extension is used by NeXTSTEP to denote RIB code files.

NeXTSTEP RIB files are displayed using the following document icon:

rib.tiff ¬

.rib Document Icon

A scene's RIB code can also be saved to an implicitly named RIB file by clicking on the "File > Save RIB" menu item,. Doing this saves the scene's RIB directly to the file that its RIB was last explicitly saved to, without bringing up the save panel. If a scene's RIB file has never been explicitly named, ThreeD prompts for a name by bringing up the save panel.

Note: ThreeD can handle RIB in only one direction Ð it can save a RIB description to a file but cannot read a RIB description from a file. If a scene is to be saved for further modification by ThreeD at a later time, it must always be saved as a `.threeD` scene file.

savingPhotoreal;¬7.2 Saving Photoreal Rendered Frames

The sequence frames of a photoreal rendered scene can be saved to a series of image files in TIFF format. ThreeD cannot modify TIFF image files Ð it is limited to saving into them and to reading them back for display only. The following sections describe saving photoreal sequences in detail.

Note: To save a photoreal rendered sequence, the image window must first be made the key window.

imageFilenames;¬7.2.1 Image Filenames

Each frame is saved to its own TIFF image file Ð one image file is created for every frame in the sequence being saved. When entering information into the save panel (described below in ;ThreeDmanual.rtfd;frameSaving;¬[7.2.3 Frame Saving]) the name of the sequence is specified.

ThreeD uses this sequence name to automatically construct names for the TIFF image files it creates. Each filename is generated by appending an underscore followed by a unique integer to the end of the sequence name. These filenames have the form:

sequenceName_integer.tiff

Filename integers start at zero for the first frame and increase by one for every subsequent frame in the sequence. For example, a three-frame sequence named "noodle" would be saved in three files named:

```
noodle_0.tiff
noodle_1.tiff
noodle_2.tiff
```

A single-frame sequence is saved in identical fashion Ð it will consist of a single file with a filename integer of zero.

ThreeD always appends a file extension of `.tiff` to the filenames Ð it is not necessary to specify an extension when entering the sequence name. The `.tiff` extension is used by NeXTSTEP to denote TIFF image files, which are displayed using the following document icon:

tiff.tiff ¬

imageCompression;¬7.2.2 Image Compression

Once a sequence name has been entered into the save panel (described below in ;ThreeDmanual.rtfd;frameSaving;¬[7.2.3 Frame Saving]), the compression value can be set. ThreeD can apply data compression to the image files it creates, thereby saving disk space. Applying compression is highly recommended, since the space savings can be significant.

The compression value is set using a control group named "Compression" located near the bottom of the save panel. This group contains three radio buttons and a slider.

Three compression options are available: none, LZW and JPEG. LZW offers a fixed, moderate-level compression with no accuracy loss, while JPEG offers a variable, high-level compression having some loss which increases with compression.

The desired compression option is selected using the radio buttons. LZW is the default. If JPEG compression is selected, its level can be set from "Lo" to "Hi" using the slider. Higher values produce higher compression ratios. The current level of JPEG compression is displayed in a text field next to the slider. A fairly modest level is used as the default.

Note that the level of JPEG compression displayed in the text field does not represent the actual compression ratio that will be obtained Ð it is only a relative indication of compression.

frameSaving;¬7.2.3 Frame Saving

Saving frames to an explicitly named sequence of image files is done by clicking on the "File > Save As..." menu item or by entering its keyboard equivalent of Command-S. Doing this brings up a custom save panel titled "Save Photoreal Frames (.tiff)". The desired directory is specified and a name for the sequence is entered in the text field labelled "Sequence Name:".

Once compression has been set or defaulted, clicking the "OK" button saves the sequence to the required number of appropriately named TIFF files in the specified directory.

If files using the same sequence name already exist in the directory, ThreeD issues an alert, warning of this fact. At this point, the save can either be proceeded with, causing the existing files to be overwritten by the new ones, or the save can be cancelled, leaving the existing files intact.

As saving progresses, ThreeD displays a circular gauge panel titled "Frames Saved" showing the number of frames saved. This number is displayed digitally and as a red sector. Once all frames have been saved, the image window title is reset to display the new sequence name.

Unlike interactive scenes, a photoreal frame sequence cannot be implicitly saved. Clicking on the "File > Save" menu item or entering its keyboard equivalent of Command-s always causes an explicit "File > Save As..." operation to be performed, bringing up the save panel for sequence name entry.

If the photoreal frame sequence has never been saved, any operation that will delete it causes ThreeD to first issue an alert, warning that the image window has been "edited" (but not saved). At this point the operation can be cancelled or proceeded with after optionally saving the sequence first.

copying;¬7.3 Copying

Scenes displayed in the interactive scene window and photoreal frames displayed in the image window can be copied to the pasteboard for subsequent pasting into other applications. Scene data cannot be pasted into ThreeD.

An interactive scene or a photoreal frame is copied by first making its associated window the key window, then clicking on the "Edit > Copy" menu item or entering the Command-c keyboard equivalent. This causes the scene or frame currently being displayed to be copied to the pasteboard.

Note that the entire contents of an interactive scene or photoreal frame window are copied Ð it is not possible to copy only parts of these windows.

Details of each type of copy are given in the following sections.

interactiveCopies;¬7.3.1 Interactive Scene Copies

Copying an interactive scene puts both a TIFF and a RIB representation of the scene onto the pasteboard. Both are therefore available for pasting into any other application supporting TIFF and/or RIB pasting.

The default opaque black of an interactive scene background is a true opaque colour. As a result, when a scene is copied and pasted into another application such as Edit, a true, black background will be displayed.

photorealCopies;¬7.3.2 Photoreal Frame Copies

Copying a photoreal frame puts a TIFF representation of the frame onto the pasteboard, making it available for pasting into any other application supporting TIFF pasting.

Unlike an interactive scene, the visually black background of a photoreal frame is not black, but transparent. As a result, when a photoreal frame is copied and pasted into another application such as Edit, the copy will retain its transparent background and show whatever opaque colour exists behind it. If pasting into

Edit, the background will be white, rather than black.

readingScenes;¬8 Reading In Scene Data

ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

ThreeD interactive scenes and photoreal rendered frames can be read in from existing files for further scene modification and frame display. The following sections describe this capability.

readingInteractive;¬8.1 Reading in Interactive Scenes

A ThreeD interactive scene can be read in from an existing scene file by clicking on the "File > Open Scene..." menu item or by entering its keyboard equivalent of

Command-o. Doing this brings up a standard open panel titled "Open Scene".

This panel displays only those files whose extension is `.threeD`, the standard extension used by ThreeD to denote scene files. The document icon used for scene files is shown in ;ThreeDmanual.rtfd;sceneFiles;¬[7.1.1 Scene Files].

The desired directory is chosen in the open panel and a scene file is selected from it. Only one scene file can be opened and read in at a time.

Clicking the panel's "OK" button then opens the selected scene file, reads the scene in from it and displays it in the interactive scene window. The window is made the key window, automatically resized if needed to the dimensions of the scene and its title is set to the scene's filename.

At this point the scene is fully reconstructed and available for further modification and photoreal rendering.

Note: As an alternative to opening a scene file from within the open panel, a `.threeD` scene file displayed with the ThreeD document icon can be directly double-clicked on instead. Doing this automatically starts up ThreeD if not already running and reads in the scene file as described above.

readingPhotoreal;¬8.2 Reading in Photoreal Rendered Frames

Photoreal rendered sequence frames can be read in from one or more existing TIFF image files by clicking on the "File > Open Images..." menu item or by entering its keyboard equivalent of Command-O. Doing this brings up a standard open panel titled "Open Images".

The open panel displays only those files whose extension is `.tiff`, the standard extension used by NeXTSTEP to denote TIFF image files. Each of these files (if created by ThreeD) contains a photoreal rendered frame and is named according to the format described in ;ThreeDmanual.rtfd;imageFilenames;¬[7.2.1 Image Filenames]. The document icon used for these files is shown in the same section.

Once the open panel has been brought up, the directory is chosen and the desired image file or files are selected from it.

Clicking the panel's "OK" button first brings up the image window and the playback control panel if not already present. The selected files are then opened and their frames are read in.

As read-in progresses, ThreeD displays a circular gauge panel titled "Images Opened" showing the number of image files opened. This number is displayed digitally and as a red sector.

Once all frames have been read in, the first (or earliest) frame in the sequence is displayed in the image window. The window is made the key window, automatically resized if needed to the dimensions of the frame and its title is set to the sequence name.

At this point the frames can be played back in the usual fashion.

imageSelection;¬8.2.1 Image File Selection

As indicated above, one or more image files in a sequence can be selected. There is no requirement that all the frames in a sequence be read in, and those that are need not form a contiguous block. It is valid to read in a subset of frames having "holes" in the sequence order.

Note, however, that regardless of the order in which files are selected in the open panel, ThreeD will always read in and display their frames in increasing numerical order, beginning with the frame from the lowest-numbered file.

All sequence image files created by ThreeD conform to the standard filename format described in ;ThreeDmanual.rtfd;imageFilenames;¬[7.2.1 Image Filenames]. This format is preferred by ThreeD when reading in frame sequences.

However, ThreeD will also accept image files whose names do not conform to the format Ð any valid TIFF file is acceptable, regardless of filename. ThreeD applies the following rules to determine the ordering of image files based on filename conformance to the standard format:

´ conforming files are set in increasing numeric order.
´ non-conforming files are set in increasing lexicographic order.
´ conforming files precede non-conforming files.

ThreeD uses the first image file in the ordered set to determine the image window's title and size according to the following rules:

´ If the name of the first image file contains an underscore character the window's title is set to the string immediately preceding the underscore.

Otherwise, the title is set to the full filename, including the `.tiff` extension.

´ The size of the window is set to the size of the frame in the first image file and remains at that size for all frames in the sequence. The window is not resized for subsequent frames, even if their sizes differ from that of the first frame.

knownProblems;¬9 Known Problems

ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

In addition to the restrictions and limitations noted throughout the preceding sections, the following problems with ThreeD are known to exist:

1. The photoreal rendering process sometimes fails to return any rendered images. This usually occurs when the rendering process is attempting to notify ThreeD that rendering is complete, but is ignored because ThreeD is busy supporting heavy interactive use.

Once this happens, further rendering attempts will likely fail as well. Rendering can be reenabled by terminating the rendering daemon process (see ;ThreeDmanual.rtfd;termAllFrames;¬[6.4.2.2 Terminating All Frames]) and/or restarting ThreeD.

2. Scenes containing many shapes at high resolution settings may cause the interactive scene display to hang, ignoring mouse actions and failing to update. The system console will show "DPS invalid port" error messages.

Deleting all shapes can sometimes clear the problem, but more usually ThreeD needs to be restarted.

3. The edit handles and axis labels of the selected shape continue to be displayed hanging in space after the shape has passed behind the camera. This is a visual anomaly only and does not materially affect the shape.

4. In certain scene configurations, rotation of the camera about its x-axis can sometimes cause it to simultaneously roll about its z-axis. The author doesn't know if this is due to program error or if it represents a true motion that would actually occur.

5. The selected shape lags its axis labels by one frame during automatic shape rotation.

6. When the root shape has no descendants and is selected using the shape hierarchy browser, the browser fails to update its display when the first two shapes are created. Correct updating begins once a third shape is created.

7. Shape rotation may halt when using very coarse drag sensitivities. Reducing the drag sensitivity corrects the problem.

8. The shape scale edit handle ignores the drag sensitivity setting.

9. During camera rotation, the camera status line shows the rotation axes to be whatever is set on the rotation axis control, even when the actual rotation is solely about the z-axis due to dragging outside the rotator circle.

10. Saving a scene containing a shape that uses the "texmap" shader with its "texname" parameter undefined (value = "none") causes ThreeD to crash. The "texname" parameter must be set to the full pathname of a texture map file. Example texture map files are located in `/NextLibrary/Textures/`and have a file extension of `.tx`.

legalNotes;¬10 Legal Notes

ÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐÐ±

The following sections describe the distribution conditions and legal disclaimers associated with ThreeD.

readingPhotoreal;¬distribution;¬10.1 Distribution

You may freely copy, distribute, and use the ThreeD program and its reference manual. All copies of the program should include the on-line help files and the info panel identifying the program and its author, and be accompanied by the complete contents of this reference manual, including these Legal Notes.

Comments can be directed by email to the author at:

`kjones@freenet.vancouver.bc.ca`

readingPhotoreal;¬disclaimers;¬10.2 Disclaimers

Kieran Jones provides ThreeD "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of ThreeD is with the user. Should ThreeD prove defective, the user assumes the cost of all necessary servicing, repair or correction.

In no event will Kieran Jones be liable to the user for damages, including any lost profits, lost monies, or other special, incidental or consequential damages arising out of the use or inability to use (including but not limited to loss of data or data being rendered inaccurate or losses sustained by third parties or a failure of the program to operate with any other programs) ThreeD, even if the user has been advised of the possibility of such damages, or for any claim by any other party.